Proceedings of the 8th International Conference on Educational Data Mining

EDM 2015

26-29 June 2015
Madrid – Spain

O.C. Santos, J.G. Boticario, C. Romero, M. Pechenizkiy,
A. Merceron, P. Mitros, J.M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz,
S. Ventura, M. Desmarais (Eds)

Olga Cristina Santos, Jesus Gonzalez Boticario, Cristobal Romero, Mykola Pechenizkiy,
Agathe Merceron, Piotr Mitros, José María Luna, Cristian Mihaescu, Pablo Moreno,
Arnon Hershkovitz, Sebastian Ventura, Michel Desmarais

International Conference on Educational Data Mining (EDM) 2015
Proceedings of the 8th International Conference on Educational Data Mining
Olga Cristina Santos, Jesus Gonzalez Boticario, Cristobal Romero, Mykola Pechenizkiy,
Agathe Merceron, Piotr Mitros, José María Luna, Cristian Mihaescu, Pablo Moreno,
Arnon Hershkovitz, Sebastian Ventura, Michel Desmarais (eds.)
Madrid, June 26-29, 2015

ISBN: 978-84-606-9425-0

Preface

The 8th International Conference on Educational Data Mining (EDM 2015) is held under auspices of the International Educational Data Mining Society at UNED, the National University for Distance Education in Spain. The conference held in Madrid, Spain, July 26-29, 2015, follows the seven previous editions (London 2014, Memphis 2013, Chania 2012, Eindhoven 2011, Pittsburgh 2010, Cordoba 2009 and Montreal 2008).

The EDM conference is a leading international forum for high-quality research that mines large data sets in order to answer educational research questions that shed light on the learning processes. These data sets may come from the traces that students leave when they interact with learning management systems, interactive learning environments, intelligent tutoring systems, educational games or when they participate in a data-rich learning context. The types of data therefore range from raw log files to eye-tracking devices and other sensor data.

This year's conference features three invited talks by Luis von Ahn and Matt Streeter (Duolingo), George Siemens, Ryan Baker and Dragan Gasevic (Athabasca University, Columbia University and University of Edinburgh respectively) and Pekka Räsänen (Niilo Mäki Institute). To facilitate further discussion of the increasingly important research issues, three interactive panels have been organized; on grand challenges in EDM, ethics and privacy considerations in EDM, and practical applications of EDM at scale. This year, together with the Journal of Educational Data Mining (JEDM), we started the JEDM Track with the intention to accommodate researchers who want to contribute a more substantial contribution than space allows in the conference proceedings, and yet to present their work to a live conference audience. The papers submitted to the track followed the regular JEDM peer review process; 4 paper have been accepted to the track and will be presented at the conference. The abstracts of the invited talks, panels and accepted JEDM Track papers can be found in these proceedings.

The main conference calls for papers invited contributions to the Research Track and Industry Track. We received 121 full and 59 short paper submissions, each of which was reviewed by three experts in the field, resulting in 43 full (41 research and 2 industry), and 50 short (46 research and 4 industry) papers accepted for presentation at the conference (some of the full paper submissions have been accepted as short paper). From a separate call for posters we also accepted 39 poster and 3 demo papers. All accepted submissions appear in these proceedings.

The EDM conference traditionally provides opportunities for young researchers, and particularly for PhD students, to present their research ideas and receive feedback from the peers and more senior researchers. This year, the organized Doctoral Consortium will feature 12 presentations.

Besides the main conference program, the participants are program conference also includes 3 workshops (Graph-based Educational Data Mining, SMLIR: Workshop on Tools and Technologies in Statistics, Machine Learning and Information Retrieval for Educational Data Mining, and International Workshop on Affect, Meta-Affect, Data and Learning) and 2 tutorials (Using Natural Language Processing Tools in Educational Data Mining, and Student Modeling Applications, Recent Developments & Toolkits).

We would like to thank UNED for the sponsorship and hosting of EDM'2015. We would like to thank the commercial sponsors (MARi, Pearson and duoLingo), student support sponsors (NSF and Professor Ram Kumar Memorial Foundation) and academic support (UNED). We also want to acknowledge the amazing work of the program committee members and additional reviewers, who with their enthusiastic contributions gave us invaluable support in putting this conference together. Our special thanks to ConferenceNavigator – a social system for conference attendees that provided services for personal scheduling, social linking and personalized recommendations of papers. Last but not least we would like to thank the local organizing team.

June 2015
Cristobal Romero and Mykola Pechenizkiy – Program Chairs
Jesus G. Boticario and Olga C. Santos – Conference Chairs

Organization

CONFERENCE CHAIRS

Jesus G. Boticario	UNED, Spain
Olga C. Santos	UNED, Spain

PROGRAM CHAIRS

Cristóbal Romero	University of Cordoba, Spain
Mykola Pechenizkiy	Eindhoven University of Technology, the Netherlands

INDUSTRY TRACK CHAIRS

Agathe Merceron	Beuth Hochschule für Technik Berlin, Germany
Piotr Mitros	edX, USA

JEDM JOURNAL TRACK CHAIR

Michel Desmarais	Ecole polytechnique Montreal, Canada

POSTER CHAIRS

José María Luna	University of Cordoba, Spain
Cristian Mihaescu	University of Craiova, Romania

DEMO CHAIRS

Pablo Moreno	e-Learning Group
Arnon Hershkovitz	Tel Aviv University, Israel

DOCTORAL CONSORTIUM CHAIRS

Sebastián Ventura	University of Cordoba, Spain

WORKSHOP AND TUTORIAL CHAIRS

Katrien Verbert	KU Leuven, Belgium
Kaska Poryaska-Pomsta	LKL

PUBLICITY CHAIRS

Sergio Gutiérrez	University of London Birkbeck, UK
Taylor Martin	NSF, USA

SPONSOR CHAIR

Dragan Gašević	University of Edinburgh, UK

AWARDS CHAIRS

Manolis Mavrikis	University of London Birkbeck, UK
Kalina Yacef	University of Sydney, Australia

ADMINISTRATIVE ORGANIZER

Mar Saneiro	UNED, Spain

LOCAL ARRANGEMENT CHAIR

Sergio Salmeron-Majadas	UNED, Spain

FINANCIAL CHAIR

Pilar Muñoz	UNED, Spain

WEB CHAIR

Emmanuelle Gutiérrez y Restrepo	UNED, Spain

STEERING COMMITTEE/IEDMS BOARD OF DIRECTORS

Ryan Baker	Teachers College, Columbia University
Tiffany Barnes	University of North Carolina at Charlotte, USA
Michel Desmarais	Ecole Polytechnique de Montreal, Canada
Sidney D'Mello	University of Notre Dame
Neil Heffernan	Worcester Polytechnic Institute, USA
Agathe Merceron	Beuth University of Applied Sciences, Germany
Mykola Pechenizkiy	Eindhoven University of Technology, Netherlands
John Stamper	Carnegie Mellon University
Kalina Yacef	The University of Sydney, Australia

PROGRAM COMMITTEE RESEARCH TRACK

Lalitha Agnihotri	McGraw Hill Education
Esma Aimeur	University of Montreal
Omar Alzoubi	School of Computer Science
Ivon Arroyo	Worcester Polytechnic Institute
Mirjam Augstein	Upper Austria University of Applied Sciences
Ryan Baker	Teachers College, Columbia University
Tiffany Barnes	North Carolina State University
Damian Bebell	Boston College
Behzad Beheshti	Ecole Polytechnique de Montreal
Gautam Biswas	Vanderbilt University
Mary Jean Blink	TutorGen, Inc.
Jesus Boticario	UNED
François Bouchet	LIP6 - Université Pierre et Marie Curie
Kristy Elizabeth Boyer	North Carolina State University
Javier Bravo Agapito	Universidad Autonoma de Madrid
Renza Campagni	Università degli Studi di Firenze
Alberto Cano	University of Cordoba
John Champaign	University of Waterloo
Min Chi	North Carolina State University
Germán Cobo	Universitat Oberta de Catalunya (UOC)
Mihaela Cocea	School of Computing, University of Portsmouth
Michel Desmarais	Ecole Polytechnique de Montreal
Sidney D'Mello	University of Notre Dame
Hendrik Drachsler	Open University of the Netherlands
Stephen Fancsali	Carnegie Learning, Inc.
Mingyu Feng	SRI International
Davide Fossati	Carnegie Mellon University in Qatar
Carlos Garcia	University of Córdoba
Dragan Gasevic	University of Edinburgh
Eva Gibaja	University of Cordoba
Daniela Godoy	ISISTAN Research Institute
Ilya Goldin	Pearson
Yue Gong	Worcester Polytechnic Institute
José González-Brenes	Pearson
Joseph Grafsgaard	North Carolina State University, Computer Science
Wilhelmiina Hamalainen	University of Eastern Finland
Neil Heffernan	Worcester Polytechnic Institute
Arnon Hershkovitz	Tel Aviv University
Roland Hubscher	Bentley University
Sébastien Iksal	University of Maine
Patrick Jermann	EPFL - Center for Digital Education
Mike Joy	University of Warwick
Kenneth Koedinger	Carnegie Mellon University
Irena Koprinska	University of Sydney
Sotiris Kotsiantis	Univerisity of Patras
Vanda Luengo	Université Joseph Fourier
Jose Maria Luna	University of Córdoba

Maria Luque	University of Córdoba
Lina Markauskaite	University of Sydney
Manolis Mavrikis	London Knowledge Lab
Oleksiy Mazhelis	University of Jyvaskyla
Riccardo Mazza	University of Lugano
Gordon McCalla	University of Saskatchewan
Victor Menendez	Universidad Autónoma de Yucatán
Agathe Merceron	Beuth University of Applied Sciences Berlin
Donatella Merlini	Università di Firenze
Cristian Mihaescu	University of Craiova
Julià Minguillón	Universitat Oberta de Catalunya
Piotr Mitros	EdX
Carlos Monroy	Rice University
Jack Mostow	Carnegie Mellon University
Tristan Nixon	Carnegie Learning Inc.
Roger Nkambou	Université du Québec À Montréal
Juan Luis Olmo	University of Córdoba
Andrew Olney	University of Memphis
Alvaro Ortigosa	Universidad Autonoma de Madrid
Abelardo Pardo	University of Sydney
Zach Pardos	UC Berkeley
Radek Pelanek	Masaryk University
Kaska Porayska-Pomsta	Birkbeck College
Martina Rau	University of Wisconsin-Madison
Steve Ritter	Carnegie Learning, Inc.
Robby Robson	Eduworks Corporation
José Raúl Romero	University of Córdoba
Carolyn Rose	Carnegie Mellon University
Olga Santos	UNED
George Siemens	SoLAR
Erica Snow	Arizona State University
John Stamper	Carnegie Mellon University
Jun-Ming Su	National Chiao Tung University
Ling Tan	Australian Council for Educational Research
Sebastián Ventura	University of Cordoba
Katrien Verbert	KU Leuven
Alina Vondavier	Educational Testing Service
Feng-Hsu Wang	Ming Chuan University
Stephan Weibelzahl	Private University of Applied Sciences Göttingen
Kalina Yacef	University of Sydney
Michael Yudelson	Carnegie Learning, Inc.
Amelia Zafra Gómez	University of Córdoba
Osmar Zaïane	University of Alberta
Alfredo Zapata González	Universidad Autonoma de Yucatan
Marta Zorrilla	University of Cantabria

PROGRAM COMMITTEE INDUSTRY TRACK

Ryan Baker	Teachers College, Columbia University
Ilya Goldin	Pearson
Michael Yudelson	Carnegie Learning, Inc.
George Siemens	University of Texas - Arlington
Steve Ritter	Carnegie Learning, Inc.
Zach Pardos	UC Berkeley
Lalitha Agnihotri	McGraw Hill Education
Alina Vondavier	Educational Testing Service (ETS)
Manolis Mavrikis	London Knowledge Lab
Mingyu Feng	SRI International
Ling Tan	Australian Council for Educational Research
Robby Robson	Eduworks Corporation
Patrick Jermann	EPFL - Center for Digital Education

ADDITIONAL REVIEWERS

Aurora Ramirez
Andreas Formiconi
Andrew Hicks
Ani Aghababyan
April Galyardt
Aurora Ramirez
Barbara Furletti
Beate Grawemeyer
Behrooz Mostafavi
Ben Toussaint
Caitlin Mills
Chen Lin
Christopher Maclellan
David García-Solórzano
Douglas Selent
Eliane Stampfer
Ekaterina Vasilyeva
Hannah Gogel
James Segedy
John Kinnebrew
Josep Grau
Juan Jesús Alcolea Picazo
Kelly Rivers
Korinn Ostrow
Kristin Stephens-Martinez
Laura Grassini
Leonardo Grilli
Linting Xue
Maria Cecilia Verri
Matthew Jacovina
Michael Sao Pedro
Mouna Selmi
Nayyar Zaidi
Nick Green
Nigel Bosch
Paolo Cintia
Peter Brusilovsky
Ran Liu
Rinat Rosenberg-Kima
Sameer Bhatnagar
Sarah Schultz
Satabdi Basu
Sébastien Lallé
Sergio Salmeron-Majadas
Seth Adjei
Shitian Shen
Sokratis Karkalas
Srecko Joksimovic
Steven Tang
Terry Peckham
Thomas Price
Vitomir Kovanovic
Yan Wang
Yun Huang
Zhongxiu Liu

Sponsors

Commercial Sponsors

Gold

https://www.mari.com/

Gold

http://researchnetwork.pearson.com/digital-data-analytics-and-adaptive-learning

Lanyard

https://www.duolingo.com/

Student Support Sponsors

http://www.nsf.gov

http://ramkumarfoundation.org/

Academic Sponsors

http://www.uned.es/

Table of Contents

Short Papers

INVITED TALKS

(abstracts)

Behind the Scenes of Duolingo

Luis Von Ahn
Duolingo and Carnegie Mellon University
biglou@duolingo.com

Matt Streeter
Duolingo
matt@duolingo.com

ABSTRACT

With over 100 million users, Duolingo is the most popular education app in the world in Android and iOS. In the first part of this talk, we will describe the motivation for creating Duolingo, its philosophy, and some of the basic techniques used to successfully teach languages and keep users engaged. The second part will focus on the machine learning and natural language processing algorithms we use to model student learning.

Personal Knowledge/Learning Graph

George Siemens
University of Texas Arlington
and
Athabasca University
gsiemens@gmail.com

Ryan Baker
Teachers College
Columbia University
baker2@exchange.tc.columbia.edu

Dragan Gasevic
Schools of Education and Informatics
University of Edinburgh
dragan.gasevic@ed.ac.uk

ABSTRACT

Educational data mining and learning analytics have to date largely focused on specific research questions that provide insight into granular interactions. These insights have been abstracted to include the development of predictive models, intelligent tutors, and adaptive learning. While there are several domains where holistic or systems models have provided additional explanatory power, work around learning has not created holistic models with the level of concreteness or richness required. The need for both granular and integrated high-level view of learning is further influenced by distributed, life long, multi-spaced learning that today defines education. Drawing on social and knowledge graph theory, we propose the development of a Personal Knowledge/Learning Graph (PKLG) - an open and learner-owned profile that addresses cognitive, affective, and related elements that reflect what a learner knows, is able to do, and processes through which she learns best. This talk will introduce PKLG, detail required technical infrastructure, and articulate how it would interact with established learning software.

Educational Neuroscience as a Tool to Understand Learning and Learning Disabilities in Mathematics

Pekka Räsänen
Niilo Mäki Institute
Jyväskylä, Finland
pekka.rasanen@nmi.fi

ABSTRACT

Becoming numerate is considered as one of the fundamental skills needed in the modern technology-driven society. The latest OECD (2013) report states that âĂIJThe way we live and work has changed profoundly âĂŞ and so has the set of skills we need to participate fully in and benefit from our hyper-connected societies and increasingly knowledge-based economies.âĂIJ The societies invest a lot on education with varying results. For some reasons there still are persons do not reach even a basic level of skills in numeracy or literacy irrespective of the recent advances in education, educational research and educational technologies.

Persons who fail in learning numeracy, even though they have had an opportunity to learn and who, based on their other skills, should have learnt, we call as having specific learning disabilities (SLD), developmental dyscalculia (DD). This discrepancy between learning opportunities, general skills and poor performance in mathematics, has intrigued researchers now more than a century. From the early beginning of the research there has been ideas that it has something to do how the brain of these persons have organized, failed to develop or damaged.

The recent developments in research methodologies, especially in brain imaging and statistical technologies, have opened new windows to analyze these brain related hypotheses. In my presentation I will open some of these windows with examples from functional brain imaging to longitudinal studies based on multivariate statistical analysis.

The new windows show different views from the DD. From one perspective the DD looks like a unitary construct with very specific symptoms in numerical processing. This view has been more typical within the brain imaging research. The other views show a complex where myriad of factors from genetic to learning experiences each contribute with a small share to the large variation of the individual skills. This view has been more typical in behavioural and cognitive studies, especially in longitudinal research. Whether a common ground can be reached, and what it needed for that, is discussed.

References

Syväoja, H., Tammelin, T., Ahonen, T. Räsänen, P., Tolvanen, A., Kankaanpää, A., and Kantomaa, M.T. (in press). Internal consistency and stability of the CANTAB neuropsychological test battery in children. Psychological Assessment.

Aunio, P. and Räsänen, P. (in press). Core numerical skills for learning arithmetic in children aged five to eight years. European Early Childhood Education Research Journal.

Räsänen, P., Käser, T., Wilson, A., von Aster, M., Maslov, A., and Maslova, U. (in press, Jan 2015). Assistive Technology for Supporting Learning Numeracy. In B O'Neill and A. Gillespie, (eds.) Assistive Technology for Cognition. Current Issues in Neuropsychology. London: Psychology Press.

Hannula-Sormunen, M. M., Lehtinen E., and Räsänen P. (in press, May 2015). Children's preschool subitizing, spontaneous focusing on numerosity and counting skills as predictors of mathematical performance 6-7 years later at school. Mathematical Thinking and Learning.

Räsänen, P. (2014). Computer-assisted Interventions on Basic Number Skills. In R. Cohen Kadosh and A. Dowker, (eds.) The Oxford Handbook of Mathematical Cognition. Oxford: Oxford University Press.

Mazzocco, M., and Räsänen, P. (2013). Contributions of longitudinal studies to evolving definitions and knowledge of developmental dyscalculia. Trends in Neuroscience and Education, 2, 65-73.

Zhang, X., Koponen, T., Räsänen, P., Aunola, K., Lerkkanen, M., and Nurmi, J. (2013). Linguistic and Spatial Skills Predict Early Arithmetic Development via Counting Sequence Knowledge. Child Development, 85(3), 1091-1107.

Price, G., Holloway, I., Räsänen, P., Vesterinen, M., and Ansari, D. (2008). Impaired parietal magnitude processing in Developmental Dyscalculia. Current Biology, 17(24).

INVITED PANELS

(abstracts)

Industry Panel: The Future of Practical Applications of EDM at Scale

Ryan Baker
Teachers College
Columbia University
baker2@exchange.tc.columbia.edu

John Carney
Carney Labs LLC
john.carney@carneylabs.com

Piotr Mitros
edX
pmitros@edx.org

Bror Saxberg
(moderator)
Kaplan Inc.
bror.saxberg@kaplan.com

John Stamper
Carnegie Mellon University
and PSLC DataShop
john@stamper.org

ABSTRACT

This mixed panel of different professionals working in EDM will be a conversation about increasing the connection between research and real-world applications. What's going on now to scale techniques for use "out there" in the field? What should researchers, funders, regulators, publishers, trainers, schools/universities and others be doing to get ready for practical work? What's in the way that we can usefully start work to address? We'll ask the audience to engage in this conversation as well - what's in your way to moving work from research environments to practically help learners at scale - and to generate more useable data at scale?

Ethics and Privacy in EDM

Dragan Gasevic
University of Edinburgh
dragan.gasevic@ed.ac.uk

Taylor Martin (moderator)
National Science Foundation
htmartin@nsf.gov

Zach Pardos
UC Berkeley
pardos@berkeley.edu

Mykola Pechenizkiy
TU Eindhoven
m.pechenizkiy@tue.nl

John Stamper
CMU and PSLC DataShop
john@stamper.org

Osmar Zaiane
University of Alberta
zaiane@ualberta.ca

ABSTRACT

Educational data mining is inherently falls into the category of the so-called secondary data analysis. It is common that data that have been collected for administrative or some other purposes at some point is considered as valuable for other (research) purpose. Collection of the student generated, student behavior and student performance related data on a massive scale in MOOCs, ITSs, LMS and other learning platforms raises various ethical and privacy concerns among researches, policy makers and the general public. This panel is aimed to discuss major challenges in ethics and privacy in EDM and how they are addressed now or should be addressed in the future to prevent any possible harm to the learners. Several experts are invited to discuss the potential and challenges of privacy-preserving EDM, ethics-aware predictive learning analytics, and availability of public benchmark datasets for EDM among others.

Grand Challenges for EDM and Related Research Areas

Ryan Baker (moderator)
Teachers College
Columbia University
baker2@exchange.tc.columbia.edu

Peter Brusilovsky (UM Inc)
School of Information Sciences
Pittsburgh University
peterb@pitt.edu

Dragan Gasevic (SoLAR)
Schools of Education and Informatics
University of Edinburgh
dragan.gasevic@ed.ac.uk

Neil T. Heffernan (AIED)
Department of Computer Science
Worcester Polytechnic Institute
nth@wpi.edu

Mykola Pechenizkiy (IEDMS)
Department of Computer Science
TU Eindhoven
m.pechenizkiy@tue.nl

Alyssa Wise (ISLS)
Faculty of Education
Simon Fraser University
alyssa_wise@sfu.ca

ABSTRACT

Educational data mining (EDM) and Learning analytics are still rather young research areas. The goal of this panel is to share different views on what major challenges researches need to address in EDM, learning analytics and related research areas including but not limited to User modeling, AI in Education, and Learning Sciences. The representatives of the corresponding communities are invited to discuss what grand challenges we should aim to address for the next five years, and which of these challenges are old and which are new, which of them peculiar to one distinct research area and which of them are shared across two or more research areas.

JEDM TRACK PAPERS

(abstracts)

Metrics for Evaluation of Student Models

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

Researchers use many different metrics for evaluation of performance of student models. The aim of this paper is to provide an overview of commonly used metrics, to discuss properties, advantages, and disadvantages of different metrics, to summarize current practice in educational data mining, and to provide guidance for evaluation of student models. In the discussion we mention the relation of metrics to parameter fitting, the impact of student models on student practice (over-practice, under-practice), and point out connections to related work on evaluation of probability forecasters in other domains. We also provide an empirical comparison of metrics. One of the conclusion of the paper is that some commonly used metrics should not be used (MAE) or should be used more critically (AUC).

Multi-Armed Bandits for Intelligent Tutoring Systems

Benjamin Clement
Inria, France
benjamin.clement@inria.fr

Didier Roy
Inria, France
didier.roy@inria.fr

Pierre-Yves Oudeyer
Inria, France
pierre-yves.oudeyer@inria.fr

Manuel Lopes
Inria, France
manuel.lopes@inria.fr

ABSTRACT

We present an approach to Intelligent Tutoring Systems which adaptively personalizes sequences of learning activities to maximize skills acquired by students, taking into account the limited time and motivational resources. At a given point in time, the system proposes to the students the activity which makes them progress faster. We introduce two algorithms that rely on the empirical estimation of the learning progress, RiARiT that uses information about the difficulty of each exercise and ZPDES that uses much less knowledge about the problem.

The system is based on the combination of three approaches. First, it leverages recent models of intrinsically motivated learning by transposing them to active teaching, relying on empirical estimation of learning progress provided by specific activities to particular students. Second, it uses state-of-the-art Multi-Arm Bandit (MAB) techniques to efficiently manage the exploration/exploitation challenge of this optimization process. Third, it leverages expert knowledge to constrain and bootstrap initial exploration of the MAB, while requiring only coarse guidance information of the expert and allowing the system to deal with didactic gaps in its knowledge. The system is evaluated in a scenario where 7–8 year old schoolchildren learn how to decompose numbers while manipulating money. Systematic experiments are presented with simulated students, followed by results of a user study across a population of 400 school children.

Developing Computational Methods to Measure and Track Learners' Spatial Reasoning in an Open-Ended Simulation

Aditi Mallavarapu
University of Illinois at Chicago
amalla5@uic.edu

Leilah Lyons
University of Illinois at Chicago
llyons@uic.edu

Tia Shelley
University of Illinois at Chicago
tshell2@uic.edu

Brian Slattery
University of Illinois at Chicago
bslatt2@uic.edu

Moira Zellner
University of Illinois at Chicago
mzellner@uic.edu

Emily Minor
University of Illinois at Chicago
eminor@uic.edu

ABSTRACT

Interactive learning environments can provide learners with opportunities to explore rich, real-world problem spaces, but the nature of these problem spaces can make assessing learner progress difficult. Such assessment can be useful for providing formative and summative feedback to the learners, to educators, and to the designers of the environments. This work adds to a growing body of research that is applying EDM techniques to more open-ended problem spaces.

The open-ended problem space under study here is an environmental science simulation. Learners were confronted with the real-world challenge of effectively placing green infrastructure in an urban neighborhood to reduce surface flooding. Learners could try out different spatial arrangements of green infrastructure and use the simulation to test each solution's impact on flooding. The learners' solutions and the solutions' performances were logged during a controlled experiment with different user interface designs for the simulation. As with many open-problem spaces, analyzing this data was difficult due to the large state space, many good solutions, and many alternate paths to those good solutions.

This work proposes a procedure for reducing the state space of solutions defined by spatial patterns while maintaining their critical spatial properties. Spatial reasoning problems are a problem class not yet examined by EDM, so this work sets the stage for further research in this area. This work also details a procedure for discovering effective spatial strategies and solution paths, and demonstrates how this information can be used to give formative feedback to the designers of the interactive learning environment.

Move your lamp post: Recent data reflects learner knowledge better than older data

April Galyardt
University of Georgia
galyardt@uga.edu

Ilya Goldin
Pearson
ilya.goldin@pearson.com

ABSTRACT

In educational technology and learning sciences, there are multiple uses for a predictive model of whether a student will perform a task correctly or not. For example, an intelligent tutoring system may use such a model to estimate whether or not a student has mastered a skill. We analyze the significance of data recency in making such predictions, i.e., asking whether relatively more recent observations of a student's performance matter more than relatively older observations. We investigate several representations of recency, such as the count of prior practice in the AFM model, and the proportion of recent successes with exponential and box kernels. We find that an exponential decay of a proportion of successes provides the summary of recent practice with the highest predictive accuracy over alternative models. As a secondary contribution, we develop a new logistic regression model, Recent-Performance Factors Analysis, that leverages this representation of recent performance, and has higher predictive accuracy than existing logistic regression models.

FULL PAPERS

Combining techniques to refine item to skills Q-matrices with a partition tree

Michel C. Desmarais
Polytechnique Montreal
michel.desmarais@polymtl.ca

Peng Xu
Polytechnique Montreal
peng.xu@polymtl.ca

Behzad Beheshti
Polytechnique Montreal
behzad.beheshti@polymtl.ca

ABSTRACT

The problem of mapping items to skills is gaining interest with the emergence of recent techniques that can use data for both defining this mapping, and for refining mappings given by experts. We investigate the problem of refining mapping from an expert by combining the output of different techniques. The combination is based on a partition tree that combines the suggested refinements of three known techniques from the literature. Each technique is given as input a Q-matrix, that maps items to skills, and student test outcome data, and outputs a modified Q-matrix that constitutes suggested improvements. We test the accuracy of the partition tree combination techniques over both synthetic and real data. The results over synthetic data show a high improvement over the best single technique with a 86% error reduction on average for four different Q-matrices. For real data, the error reduction is 55%. In addition to the substantial error reduction, the partition tree refinements provide a much more stable performance than any single technique. These results suggest that the partition tree is a valuable refinement combination approach that can effectively take advantage of the complementarity of the Q-matrix refinement techniques. It brings the goal of using a data driven approach to refine the item to skill mapping closer to real applications, although challenges remain and are discussed.

1. INTRODUCTION

Defining which skills are involved in a task is non trivial. Whereas task outcome is observable, skills are not. This layer of opacity leaves a world of possibilities to define which skills are behind task performance, and no obvious evidence to know if the proposed definition is correct or not. Means to provide such feedback would be highly valuable to teachers and designers of learning environments, and we find numerous recent efforts towards this end in the last few years. They are reviewed in section 2.

We developed an approach that takes the outpout of a combination of techniques to detect likely errors of task to skills mappings given by experts. We investigate the combination of three data-driven techniques [3, 2, 7] based on a partition tree algorithm that creates binary partitions. See also [6] for a more detailed comparison of the performance of these thee techniques.

The performance of the partition tree approach is tested over synthetic and real data. But even in the case of real data, the approach to grow the partition tree trains on synthetic performance data generated from a set of Q-matrices that are similar to the Q-matrix to refine. This procedure is chosen because only synthetic data provides a large enough training set, and because it also provides ground truth labelling of latent variables.

In the rest of this text we use the term *items* to refer to questions or tasks that can be part of a formative or summative assessment, or exercises within an e-learning environment. Skills can be the mastery of concepts, factual knowledge, or any ability involved in item outcome success. However, the models reviewed here assume a static student skills state, as opposed to the Knowledge Tracing model and its derivatives [11], for example, which rely on dynamic data. We return to this limitation in the Discussion.

The different techniques to validate a Q-matrix are first described, followed by the description of the approach, the experiments, and the results.

2. Q-MATRICES AND TECHNIQUES TO VALIDATE THEM FROM DATA

A mapping of item to skills is termed a Q-matrix. An example of a 11 items and 3 skills Q-matrix is given beside. It corresponds to the Q-matrix labelled QM 1 in the results section below. From this example, item 4 requires skill 1 only, whereas item 11 requires skills 1 and 2. If all specified skills are required to succeed the item, the Q-matrix is labeled **conjunctive**. If a any of the required skill is sufficient to the item's success, then it is labeled **disjunctive**. The **compensatory** version corresponds to the case

Q-matrix QM-1

	Skill		
Item	1	2	3
1	1	1	0
2	1	0	1
3	1	0	1
4	1	0	0
5	1	1	0
6	1	1	0
7	1	0	1
8	1	0	1
9	1	0	0
10	1	0	0
11	1	1	0

where each required item increases the chances of success in some way. Conjunctive Q-matrices the most common and all matrices of the experiments here are of this type.

The conjunctive/disjunctive distinction is also referred to as AND/OR gates. Skills models such as DINA (Deterministic Input Noisy AND) and DINO (Deterministic Input Noisy Or) make reference to this AND/OR gates terminology.

The DINA model [10] defines the probability of success to an item as a function of whether the skills required are mastered, and of two parameters, the *slip* and *guess* factors. Mastery is a binary value based on the conjunctive framework: if all required skills are mastered then the value is 1, else it is 0. Slip and guess parameters are values that generally vary on a $[0, 0.2]$ scale. The probability of success to an item j by a student i is thereby defined as:

$$P(X_{ij}=1 \mid \xi_{ij}) \;=\; (1-s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}}$$

where ξ_{ij} is 1 if student i masters all required skills of item j, 0 otherwise. s_j and g_j are the *slip* and *guess* factors.

Two techniques for Q-matrix validation surveyed here rely on the DINA model, whereas the third one relies on a matrix factorization technique called ALS (Alternative Least Squares), or more precisely ALSC for the *conjunctive* version of the technique. We briefly review each technique below.

2.1 Technique 1: MinRSS

Chiu defines a method that minimizes the residual sum of square (RSS) between the real responses and the ideal responses that follow from a given Q-matrix [2] under the DINA model. The algorithm adjusts the Q-matrix by first estimating the mastery of each student, then choosing the item with the worst RSS over to the data, and replacing it with a q-vector that has the lowest RSS, and iterates until convergence. We refer to this technique as MinRSS .

2.2 Technique 2: MaxDiff

The method defined by de la Torre [3] searches for a Q-matrix that maximizes the difference in the probabilities of a correct response to an item between examinees who possess all the skills required for a correct response to that item and examinees who do not. It also relies on the DINA model to determine item outcome probability, and on an EM algorithm to estimate the slip and guess parameters. Probability differences represents an item discrimination index: the greater the difference between the probability of a correct response given the skills required and the probability given missing skills, the greater the item is discriminant. As such, we can consider that the method finds a Q-matrix that maximizes item discrimination over all items. We refer to this technique as MaxDiff .

2.3 Technique 3: Conjunctive alternate Least-Square Factorization (ALSC)

The Conjunctive alternate Least-Square Factorization (ALSC) method is defined in [7]. Contrary to the other two methods, it does not rely on the DINA model as it has no slip and guess parameters. ALSC decomposes the results matrix $\mathbf{R}_{m\times n}$ of m items by n students as the inner product two smaller matrices:

$$\neg\mathbf{R} = \mathbf{Q}\,\neg\mathbf{S} \tag{1}$$

where $\neg\mathbf{R}$ is the negation of the results matrix (m items by n students), $\mathbf{Q}$ is the m items by k skills Q-matrix, and $\neg\mathbf{S}$ is negation of the the mastery matrix of k skills by n students (normalized for rows columns to sum to 1). By negation, we mean the 0-values are transformed to 1, and non-0-values to 0. Negation is necessary for a conjunctive Q-matrix.

The factorization consists of alternating between estimates of $\mathbf{S}$ and $\mathbf{Q}$ until convergence. Starting with the initial expert defined Q-matrix, $\mathbf{Q}_0$, a least-squares estimate of $\mathbf{S}$ is obtained:

$$\neg\hat{\mathbf{S}}_0 = (\mathbf{Q}_0^{\mathrm{T}}\,\mathbf{Q}_0)^{-1}\,\mathbf{Q}_0^{\mathrm{T}}\,\neg\mathbf{R} \tag{2}$$

Then, a new estimate of the Q-matrix, $\hat{\mathbf{Q}}_1$, is again obtained by the least-squares estimate:

$$\hat{\mathbf{Q}}_1 = \neg\mathbf{R}\,\neg\hat{\mathbf{S}}_0^{\mathrm{T}}\,(\neg\hat{\mathbf{S}}_0\,\neg\hat{\mathbf{S}}_0^{\mathrm{T}})^{-1} \tag{3}$$

And so on until convergence. Alternating between equations (2) and (3) yields progressive refinements of the matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{S}}_i$ that more closely approximate $\mathbf{R}$ in equation (1). The final $\hat{\mathbf{Q}}_\mathbf{i}$ is rounded to yield a binary matrix.

Note that $(\neg\mathbf{Q}_i^{\mathrm{T}}\,\neg\mathbf{Q}_i)$ or $(\neg\hat{\mathbf{S}}_i\,\neg\hat{\mathbf{S}}_i^{\mathrm{T}})_i$ may not be invertible, for example in the case where the matrix $\mathbf{Q}_i$ is not column full-rank, or the matrix $\mathbf{S}_i$ is not row full-rank. This is resolved by adding a very small Gaussian noise before attempting the matrix inverse.

2.4 Other techniques

We chose the three techniques described above as the candidates to combine refinements that can potentially provide more accurate suggestions than any of the individual ones, but any other equivalent technique could also be combined in the same fashion instead of the three chosen ones here. Potential candidates could be, for example, a technique based on a Bayesian approach by DeCarlo et al. [5], and recent techniques that rely on time information [13, 12]. Yet another recent approach relies item text [8] to establish the mapping of items to skills.

Although the results obtained through a combination of techniques may vary as a function of the specific techniques chosen, the general principle remains valid for all possible combinations. And there is no reason to believe that the particular combination of the current study is better or worse than other potential combinations.

2.5 General validation principle

The general idea behind the validation of Q-matrices is to introduce a perturbation to a matrix and run a refinement technique that takes the perturbed matrix and test data as input, and outputs a set of refinements. In all, 8 cases can occur and they are listed in table 1. The 8 cases are a combination of the original cell value, perturbation, and value proposed ($2 \times 2 \times 2$).

The outcome of a proposed value from the refinement technique is considered correct if it corresponds to the original value before the perturbation, and incorrect otherwise. We

Table 1: Refinement outcomes

	Perturbation		Refinement	
	Value before	Value after	Value proposed	Outcome
		Perturbed cell		
(1)	0	1	0	correct (TP)
(2)	1	0	1	correct (TP)
(3)	0	1	1	wrong (FN)
(4)	1	0	0	wrong (FN)
		Non Perturbed cell		
(5)	0	0	0	correct (TN)
(6)	1	1	1	correct (TN)
(7)	0	0	1	wrong (FP)
(8)	1	1	0	wrong (FP)

also refer to the signal detection terminology with respect to perturbations to introduce further classification of the error types:

- **True Positives (TP)**: perturbed cell that was correctly changed
- **True Negatives (TN)**: non perturbed cell left unchanged
- **False Positives (FP)**: non perturbed cell incorrectly changed
- **False Negatives (FN)**: perturbed cell left unchanged

3. COMBINING TECHNIQUES WITH A PARTITION TREE

Each of the technique described above uses a different algorithm to provide a potentially improved Q-matrix. In that respect, their respective outcome may be complementary, and their combined outcome can be more reliable than any single one. This is the first hypothesis and objective of our study. Furthermore, some algorithms are more effective in general, but may not be the best performer in all context. Identifying in which context an algorithm provides the most reliable outcome is another objective of combining these techniques. We will see that the first hypothesis is confirmed in the results of the partition tree labeled (1) and the second is also confirmed by the results of partition tree (3).

3.1 Partitioning tree

To implement the partition tree combination of the three techniques, we chose the `rpart` package for this purpose [19].

The `rpart` package builds classification models that can be represented as binary trees. The tree is constructed in a top-down recursive divide and conquer approach. At each node in the tree, cases are split into two groups based on their attribute value.

3.1.1 Tree building

Attribute selection is done on the basis of Gini index in `rpart`. The Gini index [16] can be calculated as :

$$\text{Gini}(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where n is the number of classes and p_j is the relative frequency of class j in dataset D. If attribute A is chosen to be a split on dataset D into two subset D_1 and D_2, then the Gini index for attribute A is defined as:

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2)$$

Once we get the Gini index to add attributes we can calculate a Delta reduction for each attribute:

$$\triangle\text{Gini}(A) = \text{Gini}(D) - \text{Gini}_A(D)$$

The attribute that creates the largest reduction can be chosen as a splitting point in the decision tree.

3.1.2 Classification with the tree

In our case, attributes are sometimes numeric, such as factors, and sometimes binary, such as cell values in the Q-matrix. And the class is binary since it is also a Q-matrix cell value. At each point of decision from the root node of the tree to a leaf node, a choice is made to go left or right based on the splitting point of each node. The nodes in the partition trees of this experiment are the output of the techniques (suggested values) and the factors considered (they are described in the next section).

Once a leaf node is reached, classification is based on the majority vote of the cases that fall under that leaf node: if the training set contained more case labeled '0', this is the proposed value, else it is a '1'.

3.2 Factors considered

The partition tree relies on each technique's output, the Q-matrix refinement proposition, and on a number of factors that may provide information about the most reliable technique refinement in a given context. The factors considered to be relevant are the following:

- **Skills per row**. Items can require one or more skills. The skills per row indicates the number of skills required.
- **Skills per column**. The sum of the skills per columns is an indicator of how often this skill is measured by the different items of the Q-matrix.
- **Stickiness**. If a technique systematically proposes a change to a cell of the Q-matrix, no matter what the perturbation is, this is an indication that this particular change to the original Q-matrix is an artifact of the structure of the Q-matrix and the algorithm. We call this property the *stickiness* of a cell of the matrix and it is measured by the proportion of times the value of the cell is incorrectly changed over all perturbations.

  Recall that we train the partition tree over synthetic data for which the ground truth is known. We can therefore reliably identify incorrect changes. This is detailed below.

3.3 Training of the partition tree

The partition tree is trained on data that contains the following set of attributes:

- $\texttt{original}_{(j,k)}$: value of cell (j,k) in the original matrix. This is the target class of the partition tree and it corresponds to "Value before" in table 1.
- $\texttt{MaxDiff}_{(j,k)}$, $\texttt{MinRSS}_{(j,k)}$, $\texttt{ALSC}_{(j,k)}$: the three values proposed as refinements by the respective technique in place of the original value. For every record, at least one of these must be different from the original one, or else it is a perturbated cell record. This corresponds to "Value proposed" in table 1, one for each refinement technique.
- $\texttt{RS}_{Q_i,j}$, $\texttt{CS}_{Q_i,k}$: the number of skills per row and column attributes (see section 3.2). These factors are per Q-matrix, Q_i, and per row j and column k.
- $\texttt{SF}_{\text{MaxDiff}(Q_i,j,k)}$, $\texttt{SF}_{\text{MinRSS}(Q_i,j,k)}$, $\texttt{SF}_{\text{ALSC}(Q_i,j,k)}$: the stickiness factors of the cell, one for each matrix and technique.

The training data is generated through a perturbation process. Each cell of a Q-matrix is perturbated, in turn and one at a time, to create a new training record containing the above attributes. However, non perturbated cells that are left unchanged by all refinements techniques, cases (5) and (6) in table 1, are left out of the training data because they were assumed to be uninformative.

The size of the data set to train the partition trees over is very large. For the permutations of a single Q-matrix, the number of perturbated and non perturbated cells ranges from approximately 50,000 to 250,000.

Training of the partition tree for expert Q-matrices with synthetic data. Whereas for synthetic data, we can generate a large array of Q-matrices and ample training and testing data, real data poses a challenge in that respect. Typically, for a single data set, we have only a few expert Q-matrices, and often a single one is available. For a 3 skills $\times$ 11 items matrix, only 33 single perturbations are possible to train a partition tree. Furthermore, and unlike synthetic data, we do not know what are the valid refinements in the Q-matrix. A "sticky" cell might be a valid refinement, and so can some of the perturbations that are presumed incorrect.

To get around these issues, the training of the partition tree is conducted over synthetic data where the ground truth is know and where we can use a large span of matrices similar to the expert one. Similarity to the Q-matrix to refine is achieved by random permutations the cells of the original Q-matrix. For each Q-matrix, a total of 1000 Q-matrices are generated through this permutation process. Item outcome data for 400 simulated students is also generated. The R package CDM and the `sim.din` function [15] is used for generating synthetic student item outcome data, using 0.2 slip and guess factors.

4. REAL DATA AND Q-MATRICES

The primary source of real data for our study, from which the synthetic data is also mimicked, is the well known data set on fraction algebra problems from Tatsuoka [17] (see table 1 in [4] for a description of the problems and of the skills). The data contains complete answers of 536 students to 20 questions items, but only a subset of 11 items are used by the Q-matrices in the current study. It corresponds to the set of common items to the different Q-matrices of the experiment.

Table 2: Four Q-matrices over 11 items of Tatsuoka's data set on student item outcome

	Number of skills	Number of items	Number of cases	Description
QM 1	3	11	536	Expert driven. Skill 1 shared by all items. From [9]
QM 2	5	11	536	Expert driven. From [3]
QM 3	3	11	536	Expert driven. Single skill per item. [15]
QM 4	3	11	536	Data driven, SVD-based.

The original Q-matrix of this data set contains 8 skills and, as mentioned, 20 items. However, a number of variations of this matrix have been proposed and studied with a smaller number of skills and items [9, 3, 15]. We also chose to focus on this smaller skills set since they offer three very different expert-defined Q-matrices over the same set of items. Moreover, a smaller set of skills allows us to better establish the validity of the approach on a simpler problem, leaving for later the demonstration of whether it scales correctly to larger sets. The Q-matrices are described below.

Four Q-matrices are considered. Three of them have been studied in the literature and one is defined by ourselves. Their main attributes are reported in table 2 and the actual Q-matrices are shown in figure 1 (except for QM 1 which is introduced in section 2).

	Skills of										
	QM 2					QM 3			QM 4		
Item	1	2	3	4	5	1	2	3	1	2	3
1	1	1	1	1	0	0	1	0	1	1	0
2	1	1	1	1	1	0	0	1	1	0	1
3	0	0	1	0	0	0	0	1	0	1	0
4	1	1	1	1	0	1	0	0	1	0	0
5	1	1	1	1	0	0	1	0	1	0	0
6	1	1	0	0	0	0	1	0	0	0	1
7	1	0	1	1	1	0	0	1	1	0	1
8	1	0	1	0	0	0	0	1	0	1	1
9	1	0	1	1	0	1	0	0	1	0	0
10	1	1	1	1	0	1	0	0	1	0	1
11	1	1	1	1	0	0	1	0	1	0	0

Figure 1: Q-matrices 2, 3, and 4.

As mentioned, all Q-matrices are derivatives of the Tatsuoka [17] 20 item set. QM-1, QM-2 and QM-3 are available from the CDM package. All data sets have 3 skills, except for data set 2 which has 5 skills. Data set 3 is the only one

with a single skill per item. Matrix QM 4 was created for the purpose of this study, using the three largest singular values and the items to skills **V** matrix of the SVD decomposition of the Tatsuoka data mentioned above.

Therefore, while these four Q-matrices all share the same 11 items, they vary by the number of skills, item monoticity or not, whether a skill is common to all items, and whether they are driven from data or driven from expert analysis of item skills involved.

5. GENERAL PROCEDURE SUMMARY AND METHODOLOGICAL NOTES

To ease the understanding of the general process of the experiments, and at the expense of introducing some redundancy, figure 2 summarizes the main steps and dependencies. The top greyed box illustrates the process to generate the data for partition trees training, and the synthetic data for performance evaluation. The bottom greyed box illustrates the two test procedures for real and synthetic data. We explain the figure below and fill in some details as well.

Data generation. For each of the four Q-matrices (QM_i), the data generation process (1) 1000 permutations (2). Duplicates are kept if any. For each permutation, synthetic test outcome data of 400 simulated students is created with the CDM utility `sim.din` (3). Finally, each QM is perturbated, and that Q-matrix is fed to each of the three techniques to generate training data for the partition tree described in section 3.3 (4).

Test over real data. The experiment to assess the performance over real data takes three sources of input: the Q-matrices (1), the fraction algebra data set of Tatsuoka as described in 4 (6), and finally a partition tree (5) trained from data generated (4). It outputs a set of refinements from the different partition trees and for each of the three techniques as well (7). Finally, the refinements are compared with the original Q-matrices in (1).

Test over synthetic data. For assessing the performance over synthetic data (9), the process is similar, with the main difference that refinements are based on the synthetic test outcome data generated in (3) instead of real data. And the comparison is not done over the Q-matrices in (1), but instead over the permuted Q-matrices in (2), which constitute the ground truth as they are used to generate the data.

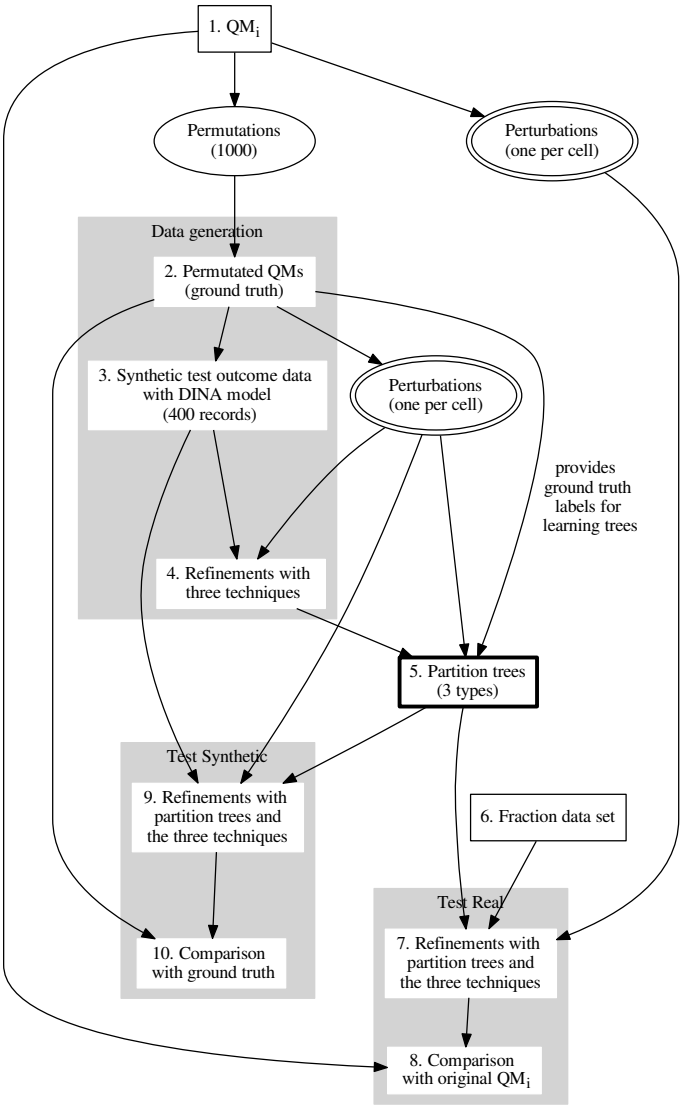


Figure 2: General validation procedure for each Q-matrix (QM_i). See section 5 for details.

5.1 Data set size, cross-validation, and the assumption of correctness of expert Q-matrices

As shown in figure 2, synthetic test outcome data (3) is used for both the training of the partition trees and testing over synthetic data. This large data set (see sect. 3.3) leaves little space for over fitting of the partition trees, and therefore the cross-validations bring very small differences in performance: accuracy/RSS error reduction is the same between a cross-validated and a non cross-validated performance assessment at the 0.01 level reported in the results below.

However, for real data, the size of the testing data set is much smaller. It varies between 366 (QM-2) and 561 (QM-3), because the test data is based solely on the permutations of the four Q-matrices. But because the test procedure uses partition trees trained from synthetic data, there are no bias issues and cross validation is not required here.

Note also that, for real data, the expert-defined Q-matrix is not necessarily consistent with the (unknown) ground truth. Nevertheless, we consider this Q-matrix as valid and the evaluation of the proposed refinements are made by comparing refinements with expert-defined Q-matrices, as though

Table 3: Results for synthetic data

QM	Technique			Partition tree		
	MinRSS	MaxDiff	ALSC	(1)	(2)	(3)
Accuracy of perturbated cells						
1	0.81	0.47	0.82	0.81	0.88	**0.95**
2	0.07	0.26	0.36	0.52	0.53	**0.83**
3	0.96	0.49	0.95	0.99	**1.00**	**1.00**
4	0.90	0.49	0.85	0.90	0.92	**0.96**
$\overline{\mathbf{X}}$	0.69	0.43	0.75	0.81	0.83	**0.93**
Accuracy of non perturbated cells						
1	0.97	0.56	0.44	0.97	0.91	**0.99**
2	**0.99**	0.53	0.50	**0.99**	**0.99**	**0.99**
3	0.95	0.26	0.74	0.95	0.94	**0.99**
4	0.97	0.56	0.44	0.97	0.97	**1.00**
$\overline{\mathbf{X}}$	0.97	0.48	0.53	0.97	0.95	**0.99**
F-score						
1	0.88	0.51	0.58	0.88	0.90	**0.97**
2	0.13	0.35	0.42	0.68	0.69	**0.90**
3	0.96	0.34	0.83	0.97	0.97	**1.00**
4	0.93	0.52	0.58	0.93	0.94	**0.98**
$\overline{\mathbf{X}}$	0.72	0.43	0.60	0.87	0.87	**0.96**

Table 4: Results for real data

QM	Technique			Partition tree		
	MinRSS	MaxDiff	ALSC	(1)	(2)	(3)
Accuracy of perturbated cells						
1	0.39	0.17	0.52	0.39	0.36	**0.67**
2	0.35	0.09	0.56	0.60	0.62	**0.64**
3	0.27	0.09	0.36	0.61	**1.00**	0.88
4	0.42	0.11	0.58	0.42	0.48	**0.61**
$\overline{\mathbf{X}}$	0.36	0.12	0.51	0.51	0.62	**0.70**
Accuracy of non perturbated cells						
1	0.45	**0.68**	0.56	0.45	0.38	0.60
2	0.93	0.93	0.28	0.94	0.94	**0.97**
3	0.64	0.83	0.42	0.69	0.76	**0.78**
4	0.55	**0.89**	0.32	0.55	0.52	0.51
$\overline{\mathbf{X}}$	0.52	0.68	0.32	0.62	0.62	**0.68**
F-score						
1	0.42	0.27	0.54	0.42	0.37	**0.63**
2	0.50	0.17	0.37	0.73	0.74	**0.77**
3	0.38	0.16	0.39	0.64	**0.86**	0.83
4	0.48	0.20	0.42	0.48	0.50	**0.56**
$\overline{\mathbf{X}}$	0.45	0.20	0.43	0.57	0.62	**0.70**

they were the ground truth. We should keep in mind that the performance score may be negatively biased if this assumption was false, but for the purpose of comparing the relative techniques performance among themselves, and if we assume that all techniques are equally affected by this bias, then it makes no difference to our relative results.

6. PERFORMANCE MEASURES

To measure the performance of the proposed refinements, we use the difference between the original Q-matrix and the proposed refinement of a technique. We use the classification of correct and incorrect refinements introduced in table 1. Cells that are neither perturbated nor incorrectly suggested as refinements by any of the technique are ignored in the analysis (the *true negatives* of table 1, TN). This is the case of the large majority and it also is consistent with the training of the partition tree for which they are also filtered out.

Recovery of a perturbated cell to its original value can be considered as a *recall* measure, whereas the non perturbated cells that are left unchanged can be considered as a *precision* measure. In that respect, we define a performance measure that combines precision and recall of the refinement technique into a single F-score measure:

$$\begin{aligned} \text{F-score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ &= 2 \times \frac{\text{Acc}_{\neg P} \times \text{Acc}_P}{\text{Acc}_{\neg P} + \text{Acc}_P} \end{aligned}$$

where Acc_P and $\text{Acc}_{\neg P}$ are respectively the accuracy measure of the proposed refinements for the perturbated and non perturbated cells. This measure gives equal weight to both types of accuracies and avoids a bias in favour of the accuracy of the non perturbated cells which can considerably outweigh in number the single perturbated cell, even after filtering out non-perturbated cells that are left unchanged.

7. RESULTS

The results are reported in tables 3 and 4. The format of these tables first described below.

7.1 Description

The respective results of the four Q-matrices (column **QM**) in table 2 are reported. They correspond to a single run (real data can vary a few percentage points by run, but it is practically stable for synthetic data due to the large number of cases). The accuracy of refinement for perturbated and non perturbated cells are reported separately, followed by the F-score which combines both types of accuracy. The averages of the four matrices for each of these these three performance measures is also reported as $\overline{\mathbf{X}}$.

The accuracy and F-score of each individual technique is reported under columns **MinRSS** , **MaxDiff** , and **ALSC**.

The three columns under **Partition tree** correspond to the performance as a function of different factors used for building the tree:

(1) MinRSS + MaxDiff + ALSC. Only the output of the three refinement techniques is considered.

(2) MinRSS + MaxDiff + ALSC + SR + SC. The number of skills per row (SR) and skills per column (SC) of the target cell are taken into account in addition to the output of each technique. If some technique performs better under some combination of SR and SC, this tree will be able to take these factor into account.

(3) MinRSS + MaxDiff + ALSC + SR + SC + Stickiness.MinRSS + Stickiness.MaxDiff + Stickiness.ALSC. The tendency of a cell to be a false positive for the MinRSS and ALSC methods are added. The Stickiness factor with MaxDiff is omitted here because it did not yield improvements.

7.2 Synthetic data

The results for synthetic data in 3 show large differences between the different matrices and across the individual techniques.

The MinRSS method is clearly superior in terms of general accuracy, except for the 5-skills Q-matrix where it can only identify the perturbated cell 7% of the time, and which brings its average below the ALSC technique. However, because it introduces fewer *false positives* (incorrect refinements) than other techniques, it outperform the other two methods on the F-Score.

On average, the ALSC technique is good at identifying the perturbated cell with a 75% average, but it also tends to introduce more false positives and consequently obtains a lower global F-score than MinRSS .

Another noticeable result is that the results for QM 3 are very good, in particular for the partition trees which have perfect performance (rounding at the second decimal). This is likely attributed to the fact that it defines a single-skill mapping.

Turning to the main questions addressed in this study, the results of partition tree (1), which uses only the three techniques' output, is equal or better on all scores than any individual one. This confirms the initial hypothesis for synthetic data. Furthermore, the inclusion of factors (partition trees (2) and (3)) also substantially improves all scores, confirming the other hypothesis that some techniques perform better under a combination of factors and that the partition tree is effectively able to take advantage of this information. The stickyness factor is by far the most effective.

7.3 Real data

The results over the real data reported in table 4 show the same trends as the synthetic data, but bring less pronounced improvements. They also support both hypothesis.

We do find an exception with the non perturbated cells where the MaxDiff accuracy is above the partition trees (1) and (2) and close to (3). This is mainly due to the fact that more "false positives" are generated by the MinRSS and ALSC techniques for real data than for synthetic data, whereas the MaxDiff technique outputs very few changes in both contexts. That observation is consistent with the results in [6].

The balance between true positives and true negatives illustrates why the F-score should be the reference: a perfect score could be obtained over the accuracy of non perturbated cells if no changes are always suggested, but that would make such refinement technique useless.

Therefore, turning to the F-scores, the tendencies are highly consistent with the synthetic data. The F-score of the best performer, 0.41 of MinRSS , is improved to 0.55 with the combination of the three techniques, and to 0.66 when all factors are included in the partition tree.

8. DISCUSSION

The results of the above experiments show that the combination of Q-matrix refinement techniques using a partition tree can bring substantial improvements over the best performance of the individual techniques. For synthetic data the average best F-score of the MinRSS technique, 0.72, is improved to 0.96, and for real data it is raised from 0.41 to 0.66. These results represent a 86% and 55% error reduction for the F-score of the synthetic and the real data respectively (error reduction $= 1 - (1 - F')/(1 - F)$, where F is the initial F-score and F' is the improved F-score).

In practical terms, if the best technique finds an error in a Q-matrix 5 out of 10 times, an error reduction of 40% represents an increase from 5, to 7 out of 10 times, and the same ratio applies to false errors reduction. And these figures rest on the assumption that we would know which technique is the best, whereas according to table 4's results the best technique varies across Q-matrices.

Another positive note on the results is that the partition tree F-scores are more stable across Q-matrices and are systematically better than any individual technique when all factors are taken into account (partition tree 3). This regularity incurs that, at least in the space of Q-matrices surveyed, one can safely choose partition tree refinements without concerns that, maybe, another technique could deliver better refinements for a specific Q-matrix.

In spite of these encouraging results, limitations and issues remain.

One limit is that the results are from a single 11 items set, and from a single domain. We can reasonably believe that the results vary across contexts and more investigation is required to assess this variability.

Another limitation is the models investigated in the current study use *static* student data: they assume that skill mastery does not change for a single student. This assumption is false for most data gathered in learning environments, where students take on exercises as they learn and are being assessed throughout the learning process. This type of data can be labeled as *dynamic* item outcome data because a student will be in different states of skills mastery as learning occurs.

In order to effectively use the existing techniques of Q-matrix refinement, we would need to be able to detect the moment when the state of skill mastery changed. Failure do do so would create noise in the data and impair the effectiveness of these techniques. Fortunately, substantial progress has been done in the recent decade or two towards detecting the moment of learning, such as the large body of work on Bayesian Knowledge Tracing and Tensor factorization (for eg. [1, 18]). We can also cite the work of [14] who refer to a time-varying skills matrix for students and test their approach on synthetic data. But apart from this recent

contribution, little work has been done on using this type of data for refining a Q-matrix, and we can only expect existing techniques to under perform with dynamic student data.

9. ACKNOWLEDGEMENTS

This research was funded under the NSERC Discovery grant of the first author.

10. REFERENCES

[1] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting learning moment-by-moment. *International Journal of Artificial Intelligence in Education*, 21(1):5–25, 2011.

[2] C.-Y. Chiu. Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, 2013.

[3] J. De La Torre. An empirically based method of Q-Matrix validation for the DINA model: Development and applications. *Journal of educational measurement*, 45(4):343–362, 2008.

[4] L. T. DeCarlo. On the analysis of fraction subtraction data: The DINA model, classification, latent class sizes, and the Q-Matrix. *Applied Psychological Measurement*, 35:8–26, 2011.

[5] L. T. DeCarlo. Recognizing uncertainty in the Q-Matrix via a bayesian extension of the DINA model. *Applied Psychological Measurement*, 36(6):447–468, 2012.

[6] M. C. Desmarais, B. Beheshti, and P. Xu. The refinement of a Q-matrix: Assessing methods to validate tasks to skills mapping. In *7th Educational Data Mining Conference*, pages 208–311, 2014.

[7] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices. In *6th International Conference, AIED 2013, Memphis, TN, USA*, pages 441–450, 2013.

[8] C. Goutte, G. Durand, and S. Léger. Towards automatic description of knowledge components. In *Proceedings of the 8th International Conference on Educational Data Mining*, page (to apppear), Madrid, Spain 2015.

[9] R. A. Henson, J. L. Templin, and J. T. Willse. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191–210, 2009.

[10] B. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[11] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. Technical report, Carnegie-Mellon University, Human Computer Interaction Institute, 2011.

[12] J. Nižnan, R. Pelánek, and J. Řihák. Mapping problems to skills combining expert opinion and student data. In *Mathematical and Engineering Methods in Computer Science*, pages 113–124. Springer, 2014.

[13] J. Nižnan, R. Pelánek, J. Řihák, et al. Using problem solving times and expert opinion to detect skills. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 433–434, London, UK 2014.

[14] S. Oeda, Y. Ito, and K. Yamanishi. Extracting latent skills from time series of asynchronous and incomplete examinations. In *Proceedings of EDM 2014, The 7th International Conference on Educational Data Mining*, pages 367–368, London, UK 2014.

[15] A. Robitzsch, T. Kiefer, A. George, A. Uenlue, and M. Robitzsch. Package CDM. 2012.

[16] L. Rokach. *Data mining with decision trees: theory and applications*. World scientific, 2007.

[17] K. Tatsuoka, U. of Illinois at Urbana-Champaign. Computer-based Education Research Laboratory, and N. I. of Education (US). *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.

[18] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In A. Verbraeck, M. Helfert, J. Cordeiro, and B. Shishkov, editors, *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011*, pages 69–78. SciTePress, 2011.

[19] T. Therneau, B. Atkinson, B. Ripley, and M. B. Ripley. Package rpart, 2014.

On the Performance Characteristics of Latent-Factor and Knowledge Tracing Models

Severin Klingler
Department of Computer Science
ETH Zurich, Switzerland
kseverin@inf.ethz.ch

Tanja Käser
Department of Computer Science
ETH Zurich, Switzerland
kaesert@inf.ethz.ch

Barbara Solenthaler
Department of Computer Science
ETH Zurich, Switzerland
sobarbar@inf.ethz.ch

Markus Gross
Department of Computer Science
ETH Zurich, Switzerland
grossm@inf.ethz.ch

ABSTRACT

Modeling student knowledge is a fundamental task of an intelligent tutoring system. A popular approach for modeling the acquisition of knowledge is Bayesian Knowledge Tracing (BKT). Various extensions to the original BKT model have been proposed, among them two novel models that unify BKT and Item Response Theory (IRT). Latent Factor Knowledge Tracing (LFKT) and Feature Aware Student knowledge Tracing (FAST) exhibit state of the art prediction accuracy. However, only few studies have analyzed the characteristics of these different models. In this paper, we therefore evaluate and compare properties of the models using synthetic data sets. We sample from a combined student model that encompasses all four models. Based on the true parameters of the data generating process, we assess model performance characteristics for over 66'000 parameter configurations and identify best and worst case performance. Using regression we analyze the influence of different sampling parameters on the performance of the models and study their robustness under different model assumption violations.

Keywords

Knowledge Tracing, Item Response Theory, synthetic data, predictive performance, robustness

1. INTRODUCTION

A fundamental part of an intelligent tutoring system (ITS) is the student model. Task selection and evaluation of the student's learning progress are based on this model, and therefore it influences the learning experience and the learning outcome of a student. Thus, accurately modeling and predicting student knowledge is essential.

Approaches for student modeling are usually based on two popular techniques: Item Response Theory (IRT) [36] and Bayesian Knowledge Tracing (BKT) [9]. The concept of IRT assumes that that the probability of a correct response to an item is a mathematical function of student and item parameters. The Additive Factors Model (AFM) [7, 8] fits a learning curve to the data by applying a logistic regression. Another technique called Performance Factors Analysis (PFA) [27] is based on the Rasch item response model [12]. BKT models student knowledge as a binary variable that can be inferred by binary observations. Performance of the original BKT model has been improved by using individualization techniques such as modeling the parameters by student and skill [23, 35, 39] or per school class [34]. Clustering approaches [25] have also proven successful in improving the prediction accuracy of BKT. Furthermore, hybrid models combining the approaches of IRT and BKT have been proposed. In [17] a dynamic mixture model has been presented to trace performance and affect simultaneously. The KT-IDEM model extends BKT by introducing item difficulty parameters [22]. Other work focused on individualizing the initial mastery probability of BKT by using IRT [38]. Logistic regression has also been used to integrate subskills into BKT [37]. Recently, two models have been introduced which synthesize IRT and BKT. Latent Factor Knowledge Tracing (LFKT) [18] individualizes the guess and slip probabilities of BKT based on student ability and item difficulty. Feature Aware Student Knowledge Tracing (FAST) [14] generalizes the individualized guess and slip probabilities to arbitrary features.

Lately, the analysis of properties of BKT has gained increasing attention. It has been shown [5] that learning BKT models exhibits fundamental identifiability problems, i.e., different model parameter estimates may lead to identical predictions about student performance. This problem was addressed by using an approach that biases the model search by Dirichlet priors to get statistically reliable improvements in predictive performance. [33] extended this work by performing a fixed point analysis of the solutions of the BKT learning task and by deriving constraints on the range of parameters that lead to unique solutions. Furthermore, it has been shown that the parameter space of BKT models

can be reduced using clustering [30]. Other research focused on analyzing convergence properties [24] of the expectation maximization algorithm (EM) for learning BKT models and exploring parameter estimates produced by EM [15]. It has been shown that convergence in the log likelihood space does not necessarily mean convergence in the parameter space. [11] have studied how good BKT is at predicting the moment of mastery. Different thresholds to assess mastery and their corresponding lag, i.e., the number of tasks that BKT needs to assess mastery (after mastery has already been achieved), have been investigated. Using multiple model fitting procedures, BKT has been compared to PFA [13]. While no differences in predictive accuracy between the models have been reported, it has been shown that for knowledge tracing EM achieves significantly higher predictive accuracy than Brute Force. Findings from other studies, however, suggest the opposite [1, 2]. In [4], upper bounds on the predictive performance have been investigated by employing various cheating models. It has been concluded that BKT and PFA perform close to these limits, suggesting that other factors such as robust learning or optimal waiting intervals should be considered to improve tutorial decision making. The predictive performance of LFKT and FAST has been compared to KT and IRT models in [19]. The evaluation is based on data from different intelligent tutoring systems.

In this work, we are interested in the properties of hybrid approaches combining latent factor and knowledge tracing models. In extension to previous work and especially to [19], we empirically evaluate the performance characteristics of the two recent hybrid models LFKT and FAST on synthetic data and compare them to the underlying approaches of BKT and IRT. We sample from a combined student model that encompasses all four models. By using synthetic data generated from the combined model, we show the robustness of the models under breaking model assumptions. By evaluating the models on 66'000 different parameter configurations we are able to rigorously explore the parameter space to demonstrate the relative performance gain between models for various regions of the parameter space. Our findings show that for the generated data sets FAST significantly outperforms all other methods for predicting the task outcome and that BKT is significantly better than FAST and LFKT at predicting the latent knowledge state. Furthermore we are able to identify the influence of different properties of a data set on model performance using regression and show best and worst case performances of the models.

2. INVESTIGATED MODELS

In an intelligent tutoring system a student is typically presented with a set of tasks to learn a specific skill. For each student n the system chooses at time t an item i from a set of items corresponding to a particular skill. The system then observes the answer $y_{n,t}$ of the student, which is assumed to be binary in this work. In the following, we briefly present four common techniques to model various latent states of the student and the tutoring environment.

BKT. Bayesian Knowledge Tracing (BKT) [9] models the knowledge acquisition of a single skill and is a special case of a Hidden Markov Model (HMM) [29]. BKT uses two latent states (*known* and *unknown*) to model if a student n has mastered a particular skill $k_{n,t}$ at time t, and two observable states (*correct* and *incorrect*) to represent the outcome of a particular task. Therefore, the probabilistic model can be fully described by a set of five probabilities. The initial probability of knowing a skill a-priori $p(k_{n,0})$ is denoted by p_I. The transition from one knowledge state $k_{n,t-1}$ to the next state $k_{n,t}$ is described by the probability p_L of transitioning from the *unknown* latent state to the *known* state and the probability p_F of transitioning from the *known* to the *unknown* state:

$$p(k_{n,t}) = k_{n,t-1}(1 - p_F) + (1 - k_{n,t-1})p_L. \quad (1)$$

In the case of BKT, p_F is fixed at 0. Finally, the task outcomes $y_{n,t}$ are modeled as

$$p(y_{n,t}) = k_{n,t}(1 - p_S) + (1 - k_{n,t})p_G, \quad (2)$$

where p_S denotes the *slip probability*, which is the probability of solving a task incorrectly despite knowing the skill, and p_G is the *guess probability*, which is the probability of correctly answering a task without having mastered the skill. Learning the parameters for a BKT model is done using maximum likelihood estimation (MLE).

IRT. Item Response Theory (IRT) [36] models the response of a student to an item as a function of latent student abilities θ_n and latent item difficulties d_i. The simplest form of an IRT model is the Rasch model, where each student n and each item i are treated independently. The outcome $y_{n,t}$ at time t is modeled using the logistic function

$$p(y_{n,t}) = \left(1 + e^{-(\theta_n - d_i)}\right)^{-1}. \quad (3)$$

A student with an ability of $\theta_n = d_i$ has a 50% chance of getting item i correct. In contrast to BKT, IRT does not model knowledge acquisition. The model parameters for the Rasch model are learned using EM.

LFKT. The Latent Factor Knowledge Tracing (LFKT) [18] model combines BKT and IRT using a hierarchical Bayesian model. On the basis of the BKT model, slip and guess probabilities are individualized based on student ability and item difficulty as

$$p_{G_{n,t}} = \left(1 + e^{-(d_i - \theta_n + \gamma_G)}\right)^{-1} \quad (4)$$

$$p_{S_{n,t}} = \left(1 + e^{-(\theta_n - d_i + \gamma_S)}\right)^{-1}, \quad (5)$$

where γ_G and γ_S are offsets for the guess and slip probabilities. The model is fit by calculating Bayesian parameter posteriors using Markov Chain Monte Carlo.

FAST. Feature Aware Student Knowledge Tracing (FAST) [14] allows for unification of BKT and IRT as well, but generalizes the individualized slip and guess probabilities to arbitrary features. Given a vector of features $\mathbf{f}_{n,t}$ for a student n at time t the adapted emission probability reads as

$$p(y_{n,t}) = \left(1 + e^{-(\boldsymbol{\omega}^T \mathbf{f}_{n,t})}\right)^{-1}, \quad (6)$$

where $\boldsymbol{\omega}$ is a vector of learned feature weights. If a set of binary indicator functions for the items and the students are used, FAST is able to represent the item difficulties d_i and student abilities θ_n from the IRT model. The parameters are fit using a variant of EM [6].

3. SYNTHETIC DATA GENERATION

Synthetic data is needed to have ground truth about the underlying data generating model, which enables the experimental evaluation of various properties of a model.

The sampling procedure starts by generating N student abilities θ_n from a normal distribution $N(0, \sigma)$. Then, it generates I item difficulties d_i from a uniform distribution $U(-\delta, \delta)$. Based on the initial probability p_I and the learn probability p_L a sequence of knowledge states $k_{n,0}, k_{n,1}, \ldots, k_{n,T}$ is sampled based on (1) and we therefore simulate data from only one skill. The time t^* at which $k_{n,t^*} = 1$ for the first time is considered as the moment of mastery. The number of sampled knowledge states is then given as $T = t^* + L$, where L denotes the lag of the simulated mastery learning system. For each student we generate a random sequence of items, i.e., item indices i. Arbitrary features from the training environment, such as answer times, help calls, problem solving strategy, engagement state of the student and gaming attempts, can have an influence on the performance of a student. To simulate those influences in a principled way, a single feature f is added to the data generating model with a varying feature weight ω (and thus varying correlation to the task outcomes $y_{n,t}$).

Based on these quantities, we sample the observations $y_{n,t}$ from a Bernoulli distribution with probability

$$p(y_{n,t}) = \left(1 + e^{-(\theta_n - d_i - \log \gamma_{n,t} + \omega f_{n,t})}\right)^{-1}, \qquad (7)$$

where

$$\gamma_{n,t} = (k_{n,t}(1 - p_S) + (1 - k_{n,t})p_G)^{-1} - 1.$$

Figure 1 gives a graphical overview of the described sampling procedure. Our sampling model has the following nine parameters: $p_I, p_L, p_S, p_G, \delta, \sigma, \omega, I, N$. The described sampling procedure allows sampling of data that exactly matches the model assumptions of all four models. To sample BKT data we set $\delta = \sigma = \omega = 0$ and (7) simplifies to the standard BKT formulation. By setting $p_S = p_G = 0.5$ and $\omega = 0$ we can sample from an IRT model. To sample from an LFKT model we set $\omega = 0$ and for FAST none of the parameters are restricted.

4. EXPERIMENTAL SETUP

Parameter space. We generated a vast number of parameter configurations in order to analyze the four models. The set of parameter configurations has been carefully designed to match real world conditions. The BKT parameters (p_I, p_G, p_S, p_L) are based on the parameter clusters found on real world data [30]. Using a normal distribution with a standard deviation of 0.02, we sampled up to 30 points (depending on the cluster size) around each cluster mean. According to common practice [16] we scaled the student abilities θ_n to have a mean of 0 and a variance of 1 and therefore $\sigma = 1$. We sampled the parameter δ (determining the range of the item difficulties) uniformly from $[0, 3]$ (according to [16]). Despite simulating only one skill, we varied the item difficulties to account for the fact that skill models tend to be imperfect in practice [7, 32, 20]. In accordance to the item difficulties, the feature weight ω was varied uniformly across $[0, 1.5]$. Feature values $f_{n,t}$ were sampled from the uniform distribution $U(-1, 1)$.

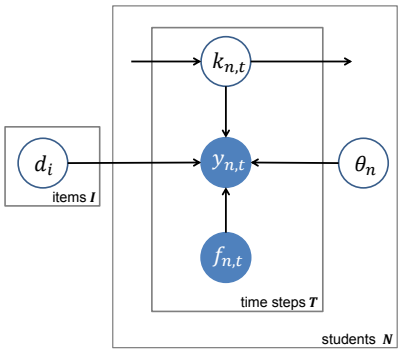


Figure 1: Combined student model used for synthetic data generation. The model corresponds to LFKT with the addition of a single feature. The relative dependencies of the observable nodes (blue) and the latent nodes (white) are shown. $k_{n,s}$ denotes the latent knowledge state, d_i the item difficulty, θ_n the student ability, $y_{n,t}$ the observation, and $f_{n,t}$ the feature value.

For every parameter configuration we generated five folds with $N = 300$ simulated students. Each fold was randomly split up into two parts of equal number of students. The first part was used as training data and the second part for testing. Therefore, the training data did contain unseen students only. As we simulated data from a mastery learning environment the number of tasks simulated for each student was determined by the moment of mastery. Based on the results presented by [11], we set the lag of the simulated system to $L = 4$ tasks from the moment of mastery. We simulated $I = 15$ different items with random item order.

In total, we generated 66′000 parameter configurations for $p_I, p_G, p_S, p_L, \delta, \omega$, this amounts total evaluation time (training and test) of 1'280 hours and 1'351 hours for LFKT and FAST respectively. The evaluation time for the BKT was 99 minutes and all configurations were evaluated in 58 minutes for the IRT model.

Implementation. To train BKT models we used our custom code that trains BKT using the Nelder-Mead simplex algorithm minimizing the log-likelihood. We thoroughly tested our implementation against the BKT implementation of [39]. The IRT models were fit by joint maximum likelihood estimation [21] implemented in the psychometrics library[1]. FAST using IRT features was shown to be equivalent to LFKT except for the parameter estimation procedure [19]. As this work did not investigate different parameter estimation techniques, both models were trained and evaluated using the publicly available FAST student modeling toolkit[2].

5. RESULTS AND DISCUSSION

Using the generated data, we investigated the performance characteristics of the four models and evaluated their predictive power and robustness under varying parameter configurations. For our results we generated 66′000 parameter

[1]An open source Java library for measurement, available at https://github.com/meyerjp3/psychometrics.
[2]http://ml-smores.github.io/fast/

configurations, and for each of them we generated synthetic data for $1'500$ students. Note that there are many ways to characterize performance differences among student models and we only cover a subset of these possibilities.

5.1 Error Metrics

The right choice of error metrics when evaluating student models has recently gained increased interest in the EDM community. In [28] some of the common error metric choices are discussed, highlighting possible issues with the accuracy and area under the ROC curve (AUC) measure. Correlations between various performance metrics and the accuracy of predicting the moment of mastering a skill has been investigated in [26], showing that the F-measure (equaling to the harmonic mean of precision and recall) and the recall are two metrics with a high correlation to the accuracy of knowledge estimation. The root mean squared error (RMSE) and log-likelihood, on the other hand, are well suited if one wants to recover the true learning parameters. Similarly, [10] concluded from results of 26 synthetic data sets that RMSE is better at fitting parameters than the log-likelihood.

In line with this previous work we investigated correlations between accuracy, RMSE and F-measure across all four models. For this, all models were trained and evaluated on data using $66'000$ different parameter configurations. All metrics are strongly correlated $|\rho| > 0.75, p \ll 0.001$. Our inspections of the metric correlations revealed no significant differences in the metric correlations among the different models. Thus, to a large extent the measures capture equal characteristics for the models we considered in this work. In the following, we therefore focus our analysis on the RMSE measure.

5.2 Model Comparison

Overall Performance. In a first step we investigated the overall performance of the models. For every parameter configuration, we calculated the average RMSE over the five generated folds. Table 1 summarizes the parameters for the best and worst data set for every model when model assumptions are met (see Section 3). Results show that all models that model a knowledge state (all except IRT) perform best if the slip probability is low and the guess probability is high. This leads to a data set that exhibits a high ratio of correct observations. IRT performs best on data that has very distinguished item difficulties (δ is high). Notably the best performance of FAST is achieved on a data set without features ($\omega = 0$). We assume that this is due to the decreased complexity of the data set, compared to one that exhibits high ω. Consistently, worst case data sets exhibit high symmetric values for guess and slip probabilities. In the case of LFTK and FAST worst case data sets additionally do not distinguish between items (difficulty range $\delta = 0$) and for FAST the feature weights are low.

We then performed the non-parametric Friedman test over all parameter configurations to assess performance differences between the models. We found that there is a statistically significant difference in the performance of the models ($\chi^2(3) = 13'065, p < 0.0001$). Performing a post-hoc analysis using Scheffe's S procedure [31] shows all model differences to be significant at $p < 0.0001$ with mean ranks of 1.7156, 2.3017, 2.6898 and 3.2929 for FAST, LFKT, BKT, and IRT, respectively. FAST therefore significantly outperforms the other methods on our synthetic data sets. In [19] IRT performed not significantly worse than LFKT and FAST on four different data sets. The good performance of IRT was attributed to the deterministic item ordering that allows IRT to infer knowledge acquisition confounded with item difficulty. Our results support this hypothesis as in our synthetic data set the items are in random order and IRT exhibits the worst overall performance.

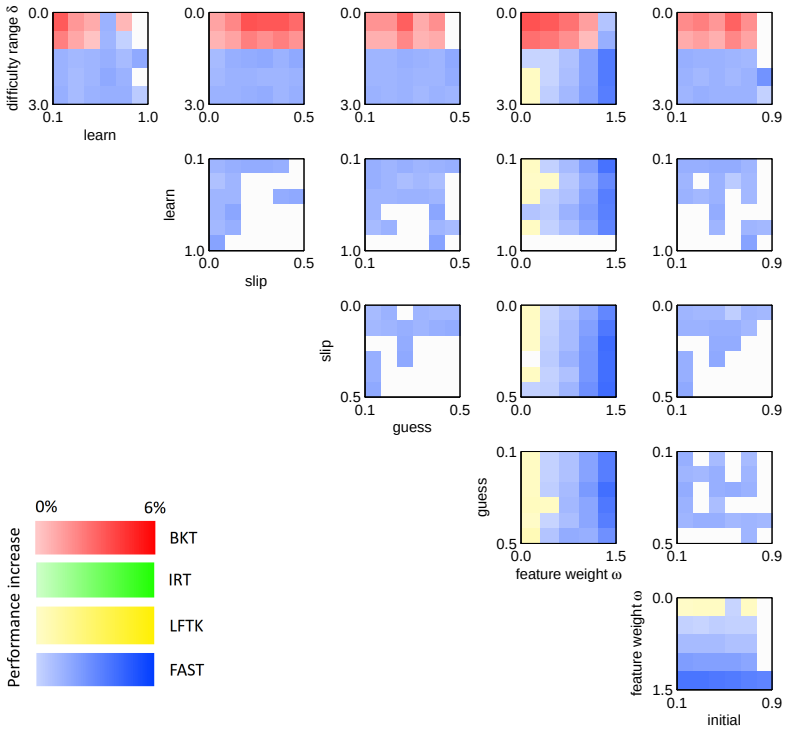
Table 1: Parameters of best and worst case data sets for each model. We only considered data sets that meet the model assumptions. Parameters denoted with * are fixed according to the model assumptions (see Section 3).

Model	δ	pI	pL	pS	pG	ω	RMSE
BKT							
Best	0.00*	0.71	0.41	0.01	0.47	0.00*	0.25
Worst	0.00*	0.10	0.12	0.50	0.49	0.00*	0.48
IRT							
Best	3.00	0.10	0.08	0.50*	0.50*	0.00*	0.42
Worst	0.00	0.10	0.10	0.50*	0.50*	0.00*	0.50
LFKT							
Best	0.75	0.69	0.40	0.01	0.46	0.00*	0.25
Worst	0.00	0.53	0.16	0.28	0.29	0.00*	0.51
FAST							
Best	0.75	0.67	0.40	0.01	0.46	0.00	0.25
Worst	0.00	0.56	0.16	0.28	0.28	0.00	0.51

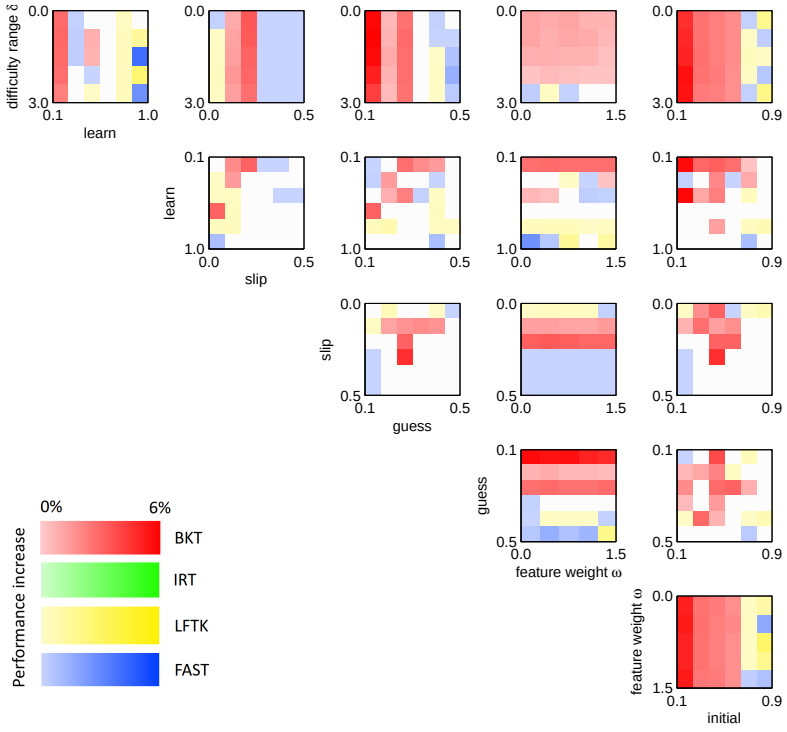
Parameter Space Investigation. To gain a better understanding of the performance characteristics of the different models, we analyzed their performances across the parameter space. For every pair of parameters p_i and p_j, we divided the parameter configurations into bins with similar values for p_i and p_j. We used five bins for each parameter (p_i and p_j) resulting in a total of 25 bins. Performance of each model was assessed by calculating the mean RMSE for each bin. Significance of the observed performance differences was computed using the Friedman test and $p < 0.05$.

Figure 2a shows the relative performance of the best model for each parameter pair. The models are color-coded: BKT is shown in red, IRT in green, LFKT in yellow, and FAST in blue. The color gradient indicates the relative improvement of the winning model over the second best model, where darker colors indicate higher values. White-colored areas indicate that there is no significant difference between the models. The plot shows that FAST is robust to parameter variations and outperforms the other models in large parts of the parameter space. In parts with low feature weights, i.e., where the feature f shows only a low correlation with task outcomes, LFKT outperforms FAST. When the variance δ of item difficulties d_i is low, BKT is the best model. A low variance in d_i implies a good skill model, with all tasks having approximately the same difficulty.

In contrast to Figure 2a, where we assessed the prediction



(a) Relative improvement in task outcome prediction (RMSE).



(b) Relative improvement in knowledge state prediction (RMSE).

Figure 2: Best performing models (RMSE) regarding prediction of task outcomes (a) and knowledge state prediction (b). The color for each bin indicates the best performing model, averaged over all other parameters. We investigated BKT (red), IRT (green), LFKT(yellow), and FAST(blue). White-colored bins exhibit no significant difference in model performance. The color brightness indicates the relative improvement of the best performing model over competing models, with dark colors referring to higher values. FAST is robust to parameter variations and outperforms the other models in large parts of the parameter space when predicting task outcomes (a). BKT is the best model if the variance of the item difficulty is low (a). BKT is superior to the other models in large parts of the parameter space when predicting knowledge states (b).

of task outcomes, we analyzed the quality of the prediction of knowledge states $k_{n,t}$ using the RMSE in Figure 2b. Ultimately, we want to predict whether a student has mastered a skill or not [26, 3]. The plot uses the same parameter pairs and color codings as Figure 2a. Interestingly, LFKT and FAST are not superior to BKT when it comes to prediction of the latent state. The additional parameters that LFKT and FAST use have a direct influence on the predicted task outcomes and therefore improve performance when predicting task outcomes. They have, however, no direct influence on the latent state $k_{n,t}$ of the model.

Robustness. Next, we tested the robustness of the different models against each other. We generated ideal data (meeting the model assumptions) for all the models and then interpolated the parameter values between these ideal cases. The classes of data sets that meet the model assumptions for the four models are described in Section 3. From every class of data sets, we selected the extreme case with the least amount of noise. In the following, we describe these cases.

For BKT, data is generated using $\delta = \omega = 0$, assuming a perfect skill model (all tasks with same difficulty) and setting the influence of additional (not captured) features to 0. Furthermore, we removed the randomness by setting $p_G = p_S = 0$. For IRT, the extreme case data was generated using $p_G, p_S = 0.5$, $\omega = 0$ and by additionally setting $\delta = 3$. As LFKT is a combination of IRT and BKT, we set the parameters to $p_G, p_S = 0.25$ and $\delta = 1.5$. Furthermore, we set $\omega = 0$, again assuming no influence of not captured features. For FAST we used the same parameters as for LFKT, but additionally introduced a feature influence by setting $\omega = 1.5$. We linearly interpolated the parameter space in-between these extreme cases to asses model robustness when model assumptions are violated. Figure 3 displays the model with best RMSE in this subspace that contains the extreme (ideal) cases, where p_L and p_I are averaged over the BKT parameter clusters presented in [30]. From these results, we can see that BKT tends to be robust to increased feature influence as long as $p_G, p_S \leq 0.15$. If the feature weight $\omega > 0.75$, FAST outperforms all the other classifiers. For large differences in item difficulties and large guess and slip probabilities, LFKT has a slight advantage over IRT.

5.3 Parameter Influence

To analyze the influence of the model parameters on the performance of the student models, we used linear regression to predict the RMSE based on the parameters of the sampling model. This allowed us to identify statistically significant correlations between the sampling parameters and the performance of the models despite the high dimensionality of the parameter space.

The sampling parameters have a direct influence on the ratio of correct observations in the data, e.g., a high learning probability with low guess and slip parameters leads to a high ratio of correct observations. Further, if the parameters model fast learners then the average number of tasks tends to be low since we are simulating a mastery learning environment. The three models IRT, LFKT and FAST which explicitly model items are sensitive to this kind of lacking data, as by having fewer observed items per student the estimation of item difficulty becomes more difficult. To

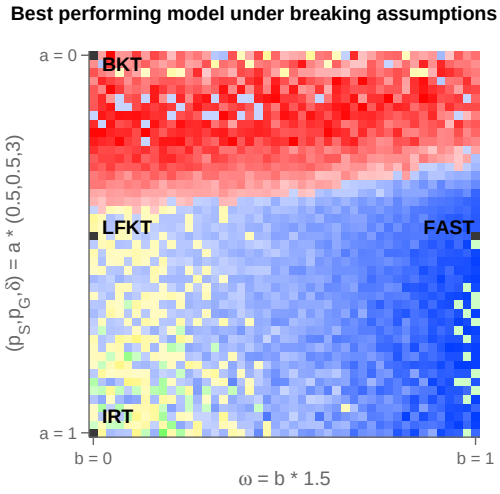


Figure 3: Relative model performance on ideal data sets generated by linearly interpolating between parameters. The colors refer to the models BKT (red), IRT (green), LFKT (yellow) and FAST (blue). The color gradient indicates the relative performance as in Figure 2a. BKT and FAST are more robust to the invalid assumptions of our experiment than IRT and LFKT.

investigate the effect of both factors, we added the two variables *correct ratio* and *average number of tasks* as predictors to the regression model. In order to make correlation coefficients comparable, all sampling parameters have been normalized to have mean 0 and standard deviation 1.

Figure 4 shows the regression coefficients for all four models, with red and green denoting statistically significant and not significant coefficients, respectively. The variables *correct ratio* and *average number of tasks* have the largest influence on the RMSE. Both effects are significant and positive (reducing the RMSE). A larger range of item difficulties δ has a positive influence on the performance of all models except for the BKT model. This is expected as BKT does not account for variations in item difficulty and thus larger variations in item difficulties are treated as noise by BKT, which makes prediction harder. IRT, LFKT and FAST, on the other hand, benefit from larger variations. We assume that this is due to the better identifiability of the effects of the different items. Interestingly, increasing the feature range ω has no significant negative effect for the models that do not take features into account (BKT, IRT, LFKT), but has a positive effect for FAST. The initial probability and the learning probability have a small negative and small positive effect on performance, respectively. While these coefficients are partially significant they have very small magnitude. The positive effect of the slip probability p_S for all models except BKT (the effect is not significant) is rather surprising. However, the effect of a high slip probability in our sampling model is that it weakens the influence of the latent knowledge state on the task outcomes. This could explain the positive influence for models that estimate item difficulty, since the difficulty estimates are less convoluted with effects from the knowledge state. Further work is needed to prove this effect.

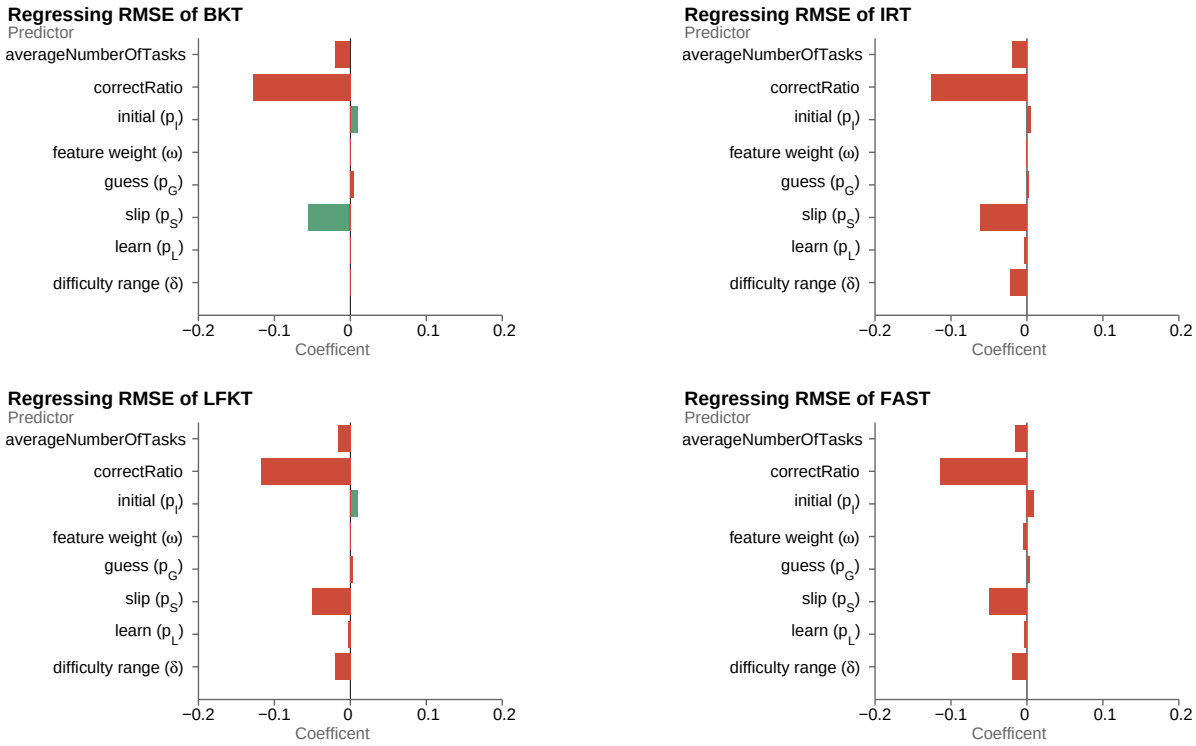


Figure 4: Regression coefficients to predict RMSE based on the sampling parameter values for the models BKT, IRT, LFKT and FAST. Parameters with positive coefficients have a negative effect on the performance and vice versa. Red denotes significant coefficients with $p < 0.001$, green coefficients are not significant.

6. CONCLUSIONS

In this work, we investigated the performance characteristics of latent factor and knowledge tracing models by exploring their parameter space. To do so, we generated a vast amount of 66'000 synthetic data sets for different parameter configurations containing data for 1'500 students each. Synthetic data allowed us to study the model performances under different parameter settings, and to test the robustness of the models against violations of specific model assumptions.

We showed best and worst case performances for all the models and investigated the relative performance gain in various regions of the parameter space. Our results showed that the two recently developed models LFKT and FAST, which synthesize item response theory and knowledge tracing, perform better than BKT and IRT. FAST even significantly outperformed LFKT if reasonable features can be extracted from the learning environment. Interestingly, IRT exhibited the worst performance, which supports the hypothesis by [19] that random item ordering has a negative influence on the performance of IRT models. However, more analyses are needed to investigate this effect thoroughly. Further, we investigated the models' abilities to predict the latent knowledge state and demonstrated that LFKT and FAST are outperformed by BKT. This raises the question of how to adjust the two recent methods LFKT and FAST if the aim is to predict knowledge states; we leave this exploration for future work. The analysis of the model robustness revealed that BKT is robust to increased feature influence for small guess and slip probabilities. For larger guess and slip, FAST outperformed the other methods.

While all sampling parameters have been carefully chosen to match real world conditions, we expect real world data to exhibit more noise and additional effects not covered by our synthetic data. Thus, the achieved performance can be considered an upper bound on the performance achievable in real world settings. The performance of BKT depends on the quality of the underlying skill model. We have simulated imperfect skill models by introducing item effects, but we did not take other sources for imperfect skill models into account. Furthermore, the simulated data consisted of a fixed set of items. For tutoring systems offering many variations of tasks, reliable estimation of item effects is challenging, which in turn influences the performance of IRT, LFKT and FAST. Moreover, the performance of FAST is driven by feature quality, which may vary between different tutoring systems.

Finally, it remains questionable whether and how the performance of the investigated techniques influences the learning outcome of students in a tutoring system. We show relative improvements in RMSE between models of up to 6%. However, the effect of small-scale improvements in the accuracy of student models on the learning outcome has been discussed controversially [4, 39].

Acknowledgments. This work was supported by ETH Research Grant ETH-23 13-2.

7. REFERENCES

[1] R. S. Baker, A. T. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual

Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In *Proc. ITS*, 2008.

[2] R. S. Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP*, 2010.

[3] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting the moment of learning. In *Proc. ITS*, 2010.

[4] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In *Proc. EDM*, 2013.

[5] J. E. Beck and K. M. Chang. Identifiability: A fundamental problem of student modeling. In *Proc. UM*, 2007.

[6] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *Proc. NAACL-HLT*, 2010.

[7] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary? - improving learning efficiency with the cognitive tutor through educational data mining. In *Proc. AIED*, 2007.

[8] H. Cen, K. R. Koedinger, and B. Junker. Comparing two IRT models for conjunctive skills. In *Proc. ITS*, 2008.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 1994.

[10] A. Dhanani, S. Y. Lee, P. Phothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in Bayesian knowledge tracing. Technical report, UCB/EECS-2014-131, EECS Department, University of California, Berkeley, 2014.

[11] S. Fancsali, T. Nixon, and S. Ritter. Optimal and worst-case performance of mastery learning assessment with Bayesian knowledge tracing. In *Proc. EDM*, 2013.

[12] G. H. Fischer and I. W. Molenaar. *Rasch models: Foundations, recent developments, and applications*. Springer Science & Business Media, 1995.

[13] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proc. ITS*, 2010.

[14] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *Proc. EDM*, 2014.

[15] J. Gu, H. Cai, and J. E. Beck. Investigate performance of expected maximization on the knowledge tracing model. In *Proc. ITS*, 2014.

[16] D. Harris. Comparison of 1-, 2-, and 3-parameter IRT models. *Educational Measurement: Issues and Practice*, 1989.

[17] J. Johns and B. Woolf. A dynamic mixture model to detect student motivation and proficiency. In *Proc. Artificial intelligence*, 2006.

[18] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proc. EDM*, 2014.

[19] M. M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. *Personalization Approaches in Learning Environments*, 2014.

[20] K. Koedinger, J. Stamper, E. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *Proc. AIED*, 2013.

[21] J. Meyer and E. Hailey. A study of Rasch, partial credit, and rating scale model parameter recovery in WINSTEPS and jMetrik. *Journal of Applied Measurement*, 2011.

[22] Z. A. Pardos and N. Heffernan. Introducing item difficulty to the knowledge tracing model. In *Proc. UMAP*, 2011.

[23] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. UMAP*, 2010.

[24] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of Bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. In *Proc. EDM*, 2010.

[25] Z. A. Pardos, S. Trivedi, N. T. Heffernan, and G. N. Sárközy. Clustered knowledge tracing. In *Proc. ITS*, 2012.

[26] Z. A. Pardos and M. Yudelson. Towards moment of learning accuracy. In *AIED Workshops*, 2013.

[27] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. In *Proc. AIED*, 2009.

[28] R. Pelánek. A brief overview of metrics for evaluation of student models. In *Approaching Twenty Years of Knowledge Tracing Workshop*, 2014.

[29] J. Reye. Student modelling based on belief networks. *IJAIED*, 2004.

[30] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. In *Proc. EDM*, 2009.

[31] H. Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.

[32] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using datashop. In *Proc. AIED*, 2011.

[33] B. van de Sande. Properties of the Bayesian knowledge tracing model. *JEDM*, 2013.

[34] Y. Wang and J. Beck. Class vs. student in a Bayesian network student model. In *Proc. AIED*, 2013.

[35] Y. Wang and N. T. Heffernan. The student skill model. In *Proc. ITS*, 2012.

[36] M. Wilson and P. De Boeck. Descriptive and explanatory item response models. 2004.

[37] Y. Xu and J. Mostow. Using logistic regression to trace multiple subskills in a dynamic Bayes net. In *Proc. EDM*, 2011.

[38] Y. Xu and J. Mostow. Using item response theory to refine knowledge tracing. In *Proc. EDM*, 2013.

[39] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian knowledge tracing models. In *Proc. AIED*, 2013.

Mixture Modeling of Individual Learning Curves

Matthew Streeter
Duolingo, Inc.
Pittsburgh, PA
matt@duolingo.com

ABSTRACT

We show that student learning can be accurately modeled using a mixture of learning curves, each of which specifies error probability as a function of time. This approach generalizes Knowledge Tracing [7], which can be viewed as a mixture model in which the learning curves are step functions. We show that this generality yields order-of-magnitude improvements in prediction accuracy on real data. Furthermore, examination of the learning curves provides actionable insights into how different segments of the student population are learning.

To make our mixture model more expressive, we allow the learning curves to be defined by generalized linear models with arbitrary features. This approach generalizes Additive Factor Models [4] and Performance Factors Analysis [16], and outperforms them on a large, real world dataset.

1. INTRODUCTION

In the mid-1980s, a now-famous study demonstrated the potential impact of adaptive, personalized education: students tutored one-on-one outperformed those taught in a conventional classroom by two standard deviations [3]. Remarkably, subsequent research has achieved similar gains using interactive, computerized tutors that maintain an accurate model of the student's knowledge and skills [6]. In the past few years, widespread access to smartphones and the web has allowed such systems to be deployed on an unprecedented scale. Duolingo's personalized language courses have enrolled over 90 million students, more than the total number of students in all U.S. elementary and secondary schools combined.

A central component of an intelligent tutoring system is the student model, which infers a student's latent skills and knowledge from observed data. To make accurate inferences from the limited data available for a particular student, one must make assumptions about how students learn. How do students differ in their learning of a particular skill or concept? Is the primary difference in the initial error rate, the rate at which error decreases with time, the shape of the learning curve, or something else? The answers to these questions have implications for the choice of model class (e.g., Hidden Markov Model, logistic regression), as well as the choice of model parameters.

Previous approaches to student modeling typically make strong assumptions about the shape of each student's learning curve (i.e., the error rate as a function of the number of trials). Additive Factor Models [4] use the student and the number of trials as features in a logistic regression model, which implies a sigmoidal learning curve with the same steepness for each student, but different horizontal offset. Knowledge Tracing [7] is a two-state Hidden Markov Model where, conditioned on the trial t at which the student first transitions from not knowing the skill to mastering it, the learning curve is a step function.

In empirical studies, it has been observed that aggregate learning curves often follow a power law, a phenomenon so ubiquitous it has been called the *power law of practice* [13]. Later work suggested that, although error rates follow a power law when averaged over an entire population, individual learning curves are more accurately modeled by exponentials [10]. That is, the power law curve observed in aggregate data is actually a mixture of exponentials, with each student's data coming from one component of the mixture.

These observations led us to seek out a more general approach to student modeling, in which individual learning curves could be teased apart from aggregate data, without making strong assumptions about the shape of the curves. Such an approach has the potential not only to make the student model more accurate, but also to explain and summarize the data in a way that can produce actionable insights into the behavior of different subsets of the student population.

This work makes several contributions to student modeling. First, we present models of student learning that generalize several prominent existing models and that outperform them on real-world datasets from Duolingo. Second, we show how our models can be used to visualize student performance in a way that gives insights into how well an intelligent tutoring system "works", improving upon the population-level learning curve analysis that is typically used for this purpose [11]. Finally, by demonstrating that relatively simple mixture models can deliver these benefits, we hope to inspire further work on more sophisticated approaches that use mixture models as a building block.

1.1 Related Work

The problem of modeling student learning is multifaceted. In full generality it entails modeling a student's latent abilities, modeling how latent abilities relate to observed performance, and modeling how abilities change over time as a result of learning and forgetting. For an overview of various approaches to student modeling, see [5, 8].

This work focuses on the important subproblem of modeling error probability as a function of trial number for a particular task. Following the influential work of Corbett and Anderson [7], Knowledge Tracing has been used to solve this problem in many intelligent tutoring systems. Recent work has sought to overcome two limitations of the basic Knowledge Tracing model: its assumption that each observed data point requires the use of a single skill, and its assumption that model parameters are the same for all students. To address the first limitation, Additive Factor Models [4] and Performance Factors Analysis [16] use logistic regressions that include parameters for each skill involved in some trial. The second limitation has been addressed by adapting the basic Knowledge Tracing model to individual students, for example by fitting per-student odds multipliers [7], or by learning per-student initial mastery probabilities [14].

Our work seeks to address a third limitation of Knowledge Tracing: its strong assumptions about the shape of the learning curve. Following Knowledge Tracing, we first attempt to model performance on a task that requires only a single skill. In §4, we generalize this approach to obtain a mixture model that includes both Additive Factor Models and Performance Factors Analysis as special cases, and that outperforms both on a large, real-world dataset.

2. SINGLE-TASK MIXTURE MODEL

In this section we present a simple mixture model that is appropriate for use on datasets with a single task. This model is a viable alternative to the basic (non-individualized) version of Knowledge Tracing, and is useful for exploratory data analysis. In §4, we generalize this model to handle datasets with multiple tasks.

2.1 The Probabilistic Model

A student's performance on a task after T trials can be represented as an error vector $v \in \{0,1\}^T$, where $v_t = 1$ if the student made an error on trial t and is 0 otherwise. Thus a task, together with a distribution over students, defines a distribution over binary error vectors. In this work, we model this distribution as a mixture of K distributions, where each component of the mixture is a *learning curve*, or equivalently a product of Bernoulli distributions (one for each trial).

To formally define this model, define the probability of observing outcome $o \in \{0,1\}$ when sampling from a Bernoulli distribution with parameter p as

$$\mathcal{B}(p,o) = \begin{cases} p & o = 1 \\ 1-p & o = 0 \ . \end{cases}$$

A learning curve $q \in [0,1]^\infty$ specifies, for each trial t, the probability q_t that the student makes an error on trial t. The probability of the error vector v according to learning curve q is $\prod_t \mathcal{B}(q_t, v_t)$. A K-component mixture over learning curves is a set $q^1, q^2, \ldots, q^K$ of learning curves, together with prior probabilities $p^1, p^2, \ldots, p^K$. The probability of an error vector $v \in \{0,1\}^T$ according to the mixture model is

$$\sum_{j=1}^{K} p^j \prod_{t=1}^{T} \mathcal{B}(q_t^j, v_t) \ .$$

Inference in a mixture model consists of applying Bayes' rule to compute a posterior distribution over the K components of the mixture, given an observed error vector. The model parameters can be fit from data using the EM algorithm, pseudo code for which is given in Algorithm 1.

Algorithm 1 EM Algorithm for single-task mixture model

Parameters: number of components K, error vector v^s for each student s, prior parameters $\alpha \geq 1$, $\beta \geq 1$.
Initialize $p^j \leftarrow \frac{1}{k}\ \forall j$, and $q_t^j \leftarrow \text{Rand}(0,1)\ \forall j,t$.
while not converged **do**
  $L_{s,j} \leftarrow p^j \prod_{t=1}^{T} \mathcal{B}(q_t^j, v_t^s)\ \forall s,j$
  $z_{s,j} \leftarrow \frac{L_{s,j}}{\sum_{j'} L_{s,j'}}\ \forall s,j$
  $q_t^j \leftarrow \frac{\alpha - 1 + \sum_s z_{s,j} v_j^s}{\alpha + \beta - 2 + \sum_s z_{s,j}}\ \forall j,t$
  $p^j \leftarrow \frac{\sum_s z_{s,j}}{\sum_s \sum_{j'} z_{s,j'}}\ \forall j$
end while

To make Algorithm 1 perform well when data is sparse, it is useful to place a Bayesian prior over the set of possible learning curves. In this work we use a product of Beta distributions for the prior: $\mathbb{P}[q] = \prod_t \text{Beta}(\alpha,\beta)(\text{q}_\text{t})$. This choice of prior gives a simple closed form for the maximization step of the EM algorithm, which can be thought of computing the maximum-likelihood estimate of q_t^j after "hallucinating" $\alpha - 1$ correct responses and $\beta - 1$ errors (see pseudo code).

2.2 Knowledge Tracing as a Mixture Model

Knowledge Tracing is typically presented as a two-state Hidden Markov Model, where the student's state indicates whether or not they have mastered a particular skill. In this section, we show that if the maximum number of trials is T, Knowledge Tracing can also be thought of as a mixture model with $T+1$ components, each of which is a step function. Thus, Knowledge Tracing can be viewed as a constrained mixture model, in contrast to the unconstrained model discussed in the previous section.

To see this relationship, recall that in a Knowledge Tracing model, the student makes an error with slip probability p_s if they have mastered the skill, and with probability $1 - p_g$ otherwise, where p_g is the probability of a correct guess. The probability of mastery is p_0 initially, and after each trial, a student who has not yet mastered the skill transitions to the mastered state with probability p_T.

Let V be an error vector, so $V_t = 1$ if the student makes an error on trial t and is 0 otherwise, and let M be the state vector: $M_t = 1$ if the student has mastered the skill at the beginning of trial t and is 0 otherwise. The distribution over

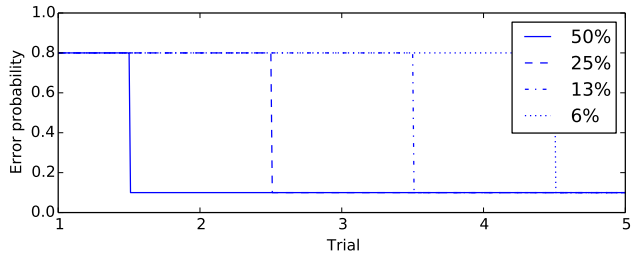


Figure 1: Mixture model representation of a Knowledge Tracing model with guess probability $p_g = 0.2$, slip probability $p_s = 0.1$, transition probability $p_T = 0.5$, and initial mastery probability $p_0 = 0$.

error vectors defined by Knowledge Tracing is given by

$$\mathbb{P}[V = v] = \sum_m \mathbb{P}[M = m]\mathbb{P}[V = v|M = m] \ .$$

Because the student never leaves the mastered state after reaching it, there are only $T + 1$ possibilities for the state vector M. Letting m^j be the jth possibility ($m^j_t = 0$ if $t < j$, 1 otherwise), and letting $p^j = \mathbb{P}[M = m^j]$, we have

$$\mathbb{P}[V = v] = \sum_{j=1}^{T+1} p^j \cdot \mathbb{P}[V = v|M = m^j] \ .$$

Because the components of V are conditionally independent given M,

$$\mathbb{P}[V = v|M = m^j] = \prod_{t=1}^{T} \mathcal{B}(q^j_t, v_t)$$

where

$$q^j_t = \begin{cases} 1 - p_g & t < j \\ p_s & t \geq j \ . \end{cases}$$

Putting these facts together, we see that the probability of a particular error vector under Knowledge Tracing is the same as under a mixture model with $T+1$ components, where each learning curve q^j is a step function with the same initial and final height but a different horizontal offset (see Figure 1).

Because the HMM and the mixture model are both generative models that specify the same distribution over binary vectors, the conditional distributions over binary vectors given a sequence of observations are also the same, and Bayesian inference yields exactly the same predictions when performed on either model.

Viewing Knowledge Tracing in this way, it is natural to consider generalizations that remove some of the constraints, for example allowing the step functions to have different initial or final heights (perhaps students who master the skill earlier are less likely to slip later on). In the model presented in §2.1 we simply remove all the constraints, allowing us to fit a mixture model over learning curves of arbitrary shape.

We note that later work on Knowledge Tracing allowed for the possibility of forgetting (transitioning from the mastered to unmastered state). This version can still be modeled as a mixture model, but with 2^T rather than $T + 1$ components.

2.3 Statistical Consistency

A model is *statistically consistent* if, given enough data, it converges to the ground truth. In this section we show that the "hard" version of EM algorithm 1 is consistent, provided the number of components in the mixture model grows with the amount of available data (the hard EM algorithm is the same as algorithm 1, except that it sets $z_{s,j} = 1$ for the j that maximizes $L_{s,j}$, and $z_{s,j} = 0$ otherwise). For simplicity we assume the number of trials T is the same for all students, but this is not essential. Also, though the data requirements suggested by this analysis are exponential T, in practice we find that near-optimal predictions are obtained using a much smaller number of components.

THEOREM 1. *Consider the "hard" version of EM algorithm 1, and suppose that the number of trials is T for all students. This algorithm is statistically consistent, provided the number of curves K in the mixture model grows as a function of the number of data points n.*

PROOF. Recall that an event occurs *with high probability* (whp) if, as $n \to \infty$, the probability of the event approaches 1. The idea of the proof is to show that, whp, each of the 2^T possible error vectors will be placed into its own cluster on the first iteration of the EM algorithm. This will imply that the EM algorithm converges on the first iteration to a mixture model that is close to the true distribution.

Consider a particular error vector $v^s \in \{0, 1\}^T$, and let j be the index of the likelihood-maximizing curve on the first iteration of the algorithm (i.e., $z_{s,j} = 1$). If $Q \in [0, 1]^T$ is a random curve, the probability that $\prod_{t=1}^{T} \mathcal{B}(Q_t, v^s_t) > \frac{1}{2}$ is positive. Thus, as $K \to \infty$, whp at least one of the K random curves will satisfy this inequality, and in particular for the likelihood-maximizing curve q^j we have $\prod_{t=1}^{T} \mathcal{B}(q^j_t, v^s_t) > \frac{1}{2}$, which implies $\mathcal{B}(q^j_t, v^s_t) > \frac{1}{2}$ for all t. For any error vector $v^{s'} \neq v^s$, there must be some t such that $v^s_t \neq v^{s'}_t$, which implies $\mathcal{B}(q^j_t, v^{s'}_t) < \frac{1}{2}$. This means that whp, q^j cannot be the likelihood-maximizing curve for $v^{s'}$, and so each binary vector will have a unique likelihood-maximizing curve.

If each binary vector v has a unique likelihood-maximizing curve q^j, then the M step of the algorithm will simply set $q^j \leftarrow v$, and will set p^j to the empirical frequency of v within the dataset. As $n \to \infty$, this empirical frequency approaches the true probability, which shows that the algorithm is consistent. □

In the worst case, statistical consistency requires a constant amount of data for every possible error vector, hence the data requirements grow exponentially with T. However, this is not as bad as it may seem. In intelligent tutoring systems, it is often the case that T is small enough that even in the worst case we can guarantee near-optimal performance. Furthermore, as we show experimentally in §3.2, near-optimal performance can often be achieved with a much smaller number of components in practice.

2.4 Use in an Intelligent Tutoring System

How should the predictions of a mixture model be used to schedule practice within an intelligent tutoring system? When using Knowledge Tracing, a typical approach is to schedule practice for a skill until the inferred probability of having mastered it exceeds some threshold such as 0.95. With a mixture model, we can no longer take this approach since we don't make explicit predictions about whether the student has mastered a skill. Nevertheless, we can define a reasonable practice scheduling rule in terms of predicted future performance.

In particular, note that another way of formulating the scheduling rule typically used in Knowledge Tracing is to say that we stop practice once we are 95% confident that performance has reached an asymptote. With a mixture model, it is unlikely that the marginal value of practice will be exactly 0, so this precise rule is unlikely to work well (it would simply schedule indefinite practice). However, we can compute the expected marginal benefit of practice (in terms of reduction in error rate), and stop scheduling practice once this drops below some threshold.

Note that when practice scheduling is defined in terms of expected marginal benefit, the practice schedule is a function of the predicted distribution over error vectors, so mixture models that make the same predictions will result in the same practice schedule even if the model parameters are different. This is in contrast to Knowledge Tracing, where multiple globally optimal models (in terms of likelihood) can lead to very different practice schedules, because the inferred probability of mastery can be different even for two models that make identical predictions [2].

2.5 Identifiability

A statistical model is identifiable if there is a unique set of parameters that maximize likelihood. Our mixture model is not identifiable, since in general there are many ways to express a given distribution over binary vectors as a mixture of learning curves. However, as we argued in the previous section, non-identifiability does not pose a problem for practice scheduling if the schedule is defined in terms of the model's predictions rather than its parameters.

3. EXPERIMENTS WITH SINGLE-TASK MODEL

In this section we evaluate the single-task mixture model of §2 on data from Duolingo. These experiments serve two purposes. First, they show that the mixture model can give much more accurate predictions than Knowledge Tracing on real data. Second, inspection of the learning curves produced by the mixture model reveals interesting facts about the student population that are not apparent from conventional learning curve analysis. In §4 we present a more general mixture model that is appropriate for datasets with multiple skills.

3.1 The Duolingo Dataset

We collected log data from Duolingo, a free language learning application with over 90 million students. Students who use Duolingo progress through a sequence of lessons, each of which takes a few minutes to complete and teaches certain words and grammatical concepts. Within each lesson, the student is asked to solve a sequence of self-contained challenges, which can be of various types. For example, a student learning Spanish may be asked to translate a Spanish sentence into English, or to determine which of several possible translations of an English sentence into Spanish is correct.

For these experiments, we focus on *listen challenges*, in which the student listens to a recording of a sentence spoken in the language they are learning, then types what they hear. Listen challenges are attractive because, unlike challenges which involve translating a sentence, there is only one correct answer, which simplifies error attribution. For these experiments we use a simple bag-of-words knowledge component (KC) model. There is one KC for each word in the correct answer, and a KC is marked correct if it appears among the words the student typed. For example, if a student learning English hears the spoken sentence "I have a business card" and types "I have a business car", we would mark the KC *card* as incorrect, while marking the KCs for the other four words correct. This approach is not perfect because it ignores word order as well as the effects of context (students may be able to infer which word is being said from context clues, even if they cannot in general recognize the word when spoken). However, the learning curves generated by this KC model are smooth and monotonically decreasing, suggesting that it performs reasonably well.

Our experiments use data from the Spanish course for English speakers, one of the most popular courses on Duolingo. In this section, we focus on modeling acquisition of a single skill, using data for the KC *una* (the feminine version of the indefinite article "a"). In §4 we consider more general mixture models, and in §5 we evaluate them on datasets with multiple KCs. The full dataset has roughly 700,000 data points (there is one data point for each combination of student, trial, and KC), while the *una* dataset contains around 15,000.

3.2 Prediction Accuracy

To evaluate the mixture model's prediction accuracy, we divided the Duolingo dataset into equal-sized training and test sets by assigning each student to one of the two groups at random. We then ran the EM algorithm on the training data to fit mixture models with various numbers of components, as well as a Knowledge Tracing model, and computed the predictions of these models on the test data. We evaluate prediction accuracy using two commonly-used metrics.

1. *Average log-likelihood.* Log-likelihood measures how probable the test data is according to the model. Specifically, if the dataset D consists of n independent data points $D_1, D_2, \ldots, D_n$ (each data point is the binary performance of a particular student on a particular trial), and $p_i = \mathbb{P}[D_i|M]$ is the conditional probability of the ith data point D_i given the model M, then

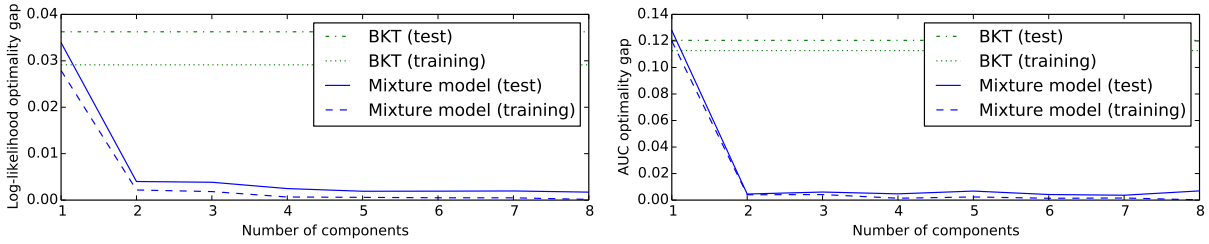


Figure 2: Optimality gaps for log likelihood (left) and AUC (right) as a function of number of components in the mixture model, compared to Knowledge Tracing (horizontal lines). The optimality gap is the absolute difference between the model's accuracy and the maximum possible accuracy on the dataset.

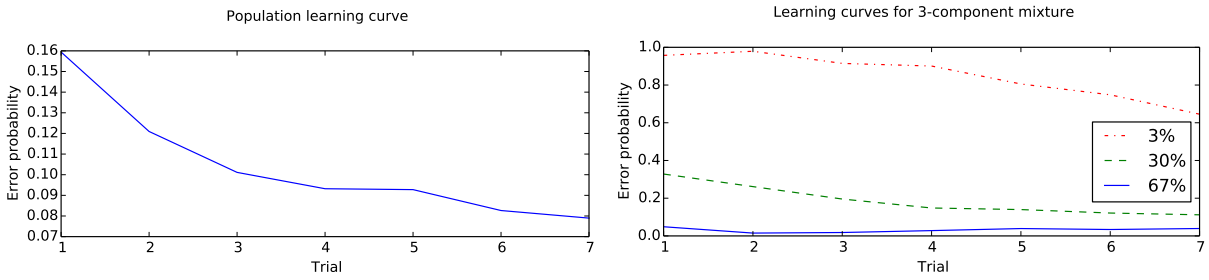


Figure 3: Learning curves for recognizing the Spanish word *una* in a Duolingo listen challenge. The population curve (left) suggests a reasonable rate of learning in aggregate, but the mixture model (right) reveals large differences among different clusters of students.

average log-likelihood is

$$\frac{1}{n}\log \mathbb{P}[D|M] = \frac{1}{n}\log \prod_{i=1}^{n} p_i = \frac{1}{n}\sum_{i=1}^{n} \log p_i \ .$$

Because both the mixture model and Knowledge Tracing are fit using maximum likelihood, it is natural to compare them in terms of this objective function.

2. *AUC.* AUC evaluates the accuracy of the model's predictions when they are converted from probabilities to binary values by applying a threshold. It can be defined as the probability that $p > q$, where p is the model's prediction for a randomly-selected positive example and q is the model's prediction for a randomly-selected negative example. This is equivalent to the area under the ROC curve, which plots true positive rate against false positive rate (both of which vary as a function of the chosen threshold).

Figure 2 presents accuracy on the *una* dataset as a function of the number of components in the mixture model, both on training and held-out test data. To make relative improvements clearer, we plot the optimality gap rather than the raw value of the prediction accuracy metric. For example, the optimality gap for test set log likelihood is the difference between the optimal log likelihood on the test data (which can be computed in closed form) and the model's log likelihood on the test data.

For both AUC and log-likelihood, the improvement in accuracy is largest when going from one component to two, and there are diminishing returns to additional components, particularly in terms of performance on held-out test data. With more than 5 components, log-likelihood on test data gets slightly worse due to overfitting, while performance on training data improves slightly. In practice, the number of components can be selected using cross-validation.

For both metrics, Knowledge Tracing is similar to the one-component model but significantly worse than the two component model in terms of accuracy, both on training and test data. Furthermore, all mixture models with two or more components outperform Knowledge Tracing by an *order of magnitude* in terms of the optimality gap for log-likelihood and AUC, both on training and on held-out test data. We observed very similar results for datasets based on other Spanish words, such as *come* (eat), *mujer* (woman), and *hombre* (man).

3.3 Learning Curve Mixture Analysis

In this section we examine the learning curves that make up the components of the mixture model fit to Duolingo data. This analysis can be viewed as a more general version of learning curve analysis [11], which examines the population learning curve (this is equivalent to the curve for a one-component mixture model).

Figure 3 presents learning curves for the *una* dataset. The left pane of the figure shows the aggregate learning curve, while the right pane shows the curves for a 3-component mixture model fit using the EM algorithm. Examining the right pane, we see that the mixture model clusters students into three quite different groups.

- Around two-thirds of the students belong to a cluster that in aggregate has an error probability around 5% on the first trial, and this error rate does not change with increased trials.
- A second, smaller cluster contains 30% of the students. These students, in aggregate, have an initial error rate of 33% which decreases to around 11% after 7 trials.
- The third cluster contains only 3% of students. These students have a very high initial error rate of 96%, which declines to about 65% after 7 trials.

The existence of this third, high-error-rate cluster surprised us, so we went back to the log data to examine the behavior of students in this cluster in more detail. It turned out that almost all of these students were simply giving up when presented with a listen challenge (although they correctly answered other types of challenges). Further examination of the log data revealed that some of these students skipped all listen challenges, while others would skip all listen challenges for long stretches of time, then at other times would correctly answer listen challenges. We conjecture that the former set of students are either hearing-impaired or do not have working speakers, while the latter do not want to turn their speakers on at certain times, for example because they are in a public place. Duolingo attempts to accommodate such students by offering a setting that disables listen challenges, but not all students realize this is available. As a result of these insights, Duolingo is now exploring user interface changes that will actively detect students that fall into this cluster and make it easier for them to temporarily disable listen challenges.

This analysis shows how mixture modeling can produce valuable insights that are not apparent from examination of the population learning curve alone. We hope this will inspire the use of mixture modeling more broadly as a general-purpose diagnostic tool for intelligent tutoring systems.

4. GENERAL MIXTURE MODEL

The single-task model is appropriate for datasets where there is a single knowledge component (KC) and many students. In an actual intelligent tutoring system, a student will learn many KCs, and prediction accuracy can improved by using student performance on one KC to help predict performance on other, not yet seen KCs. In this section we present a more general mixture model that accomplishes this.

In this more general model, student performance is again modeled as a mixture of K learning curves. However, instead of treating each point on the learning curve as a separate parameter, we let it be the output of a generalized linear model with features that depend on the student, task, and trial number. In particular, for a student s and task i, the probability of a performance vector $v_1, v_2, \ldots, v_T$ is

$$\sum_{j=1}^{k} p^j \prod_{t=1}^{T} \mathcal{B}(q^j(s,i,t;\beta^j), v_t)$$

where

$$q^j(s,i,t;\beta^j) = g^{-1}(\phi_{s,i,t} \cdot \beta^j),$$

where $\phi_{s,i,t}$ is the feature vector for student s, task i, trial t, and g is the link function for the generalized linear model [12]. Our experiments use logistic regression, for which the link function is $g(p) = \text{logit}(p)$.

Note that this model generalizes the single-task mixture model presented in §2. In particular, the single-task model with curve $q^j(t)$ is recovered by setting $\phi_{s,i,t} = e_t$, an indicator vector for trial t, and setting $\beta^j_t = g(q_j(t))$.

As with the single-task model, we can estimate the parameters of this model using the EM algorithm. The main difference is that the maximization step no longer has a closed form solution. However, it is a convex optimization and can still be solved exactly using a number of algorithms, for example stochastic gradient descent.

To define the EM algorithm, first define the likelihood function

$$L^j_{s,i}(\beta) = \prod_{t=1}^{T} \mathcal{B}(q^j(s,i,t;\beta), v_t) \ .$$

For the E step, we define hidden variables $z^j_{s,i}$, which give the probability that the data for student s and task i follows curve j.

$$z^j_{s,i} = \frac{p^j L^j_{s,i}}{\sum_{j'} p^{j'} L^{j'}_{s,i}(\beta)} \ .$$

For the M step, we optimize the coefficient vector for each component j so as to maximize expected log-likelihood.

$$\beta^j = \text{argmax}_\beta \left\{ \sum_s \sum_i z^j_{s,i} \log(L^j_{s,i}(\beta)) \right\} \ .$$

When performing inference for a new student, we solve a similar optimization problem, but we only update the coefficients for that particular student.

4.1 Relationship to Other Models

This mixture model is quite general, and with appropriate choices for the feature function ϕ can recover many previously-studied models. In particular, any modeling approach that is based on a logistic regression using features that depend only on the student, task, and trial number can be recovered by using a single component ($K = 1$), choosing g = logit, and defining ϕ to include the appropriate features. This includes both Additive Factor Models [4] and Performance Factors Analysis [16]. By choosing a larger K, we immediately obtain generalizations of each of these methods that have the potential to more accurately model the behavior of individual clusters of students. Because the trial number (together with the student and task) identifies a unique learning event, we can also include features that depend on the trial type, elapsed time, and previous learning history, as in learning decomposition [1].

Note that for the mixture model to add value over a simple regression, we must define "task" in such a way that we observe multiple trials for a given (student, task) pair. For datasets where each item requires the use of multiple KCs,

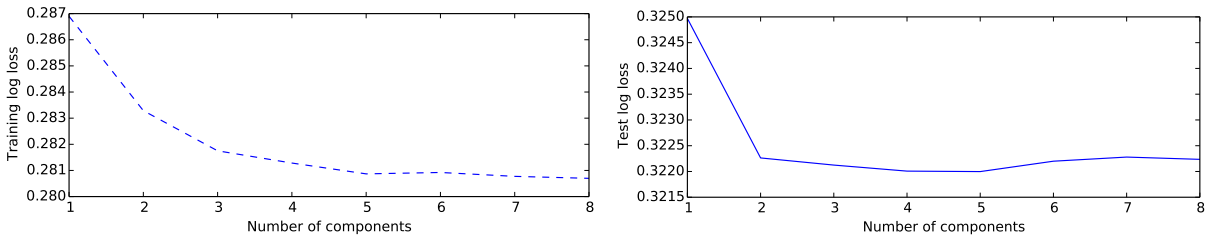


Figure 4: Performance of a mixture of Additive Factor Models on training data (left) and test data (right), as a function of the number of components in the mixture model.

Table 1: Performance on Duolingo dataset

Method	Training log loss	Test log loss	Training AUC loss	Test AUC loss
Knowledge Tracing	0.3429	0.3441	0.3406	0.3460
Performance Factors Analysis	0.3248	0.3285	0.2774	0.2865
Additive Factor Model	0.2869	0.3250	0.1629	0.2789
A.F.M. Mixture (3 components)	**0.2818**	**0.3220**	**0.1598**	**0.2760**

this entails either (a) defining a task for each combination of KCs, or (b) using error attribution to create a dataset in which each example involves only a single KC, and having one task per KC. We use the latter approach in our experiments in §5. This approach is different from the one taken by algorithms such as LR-DBN [17], which make predictions on multiple-KC items directly.

4.2 Parameter Sharing

To make more efficient use of available data when fitting this generalized mixture model, it can be useful for certain coefficient values to be shared across components of the mixture model. To illustrate this issue, consider fitting a mixture of Additive Factor Models. In this case, ϕ includes an indicator feature for each student. If we fit a K component mixture, we must estimate K separate coefficient values for each student, which increases the variance of the estimates compared to the basic Additive Factor Model. For students for whom we do not yet have much data, this can result in larger values of K giving worse performance.

To overcome this difficulty, we allow certain coefficients to be shared across all components of the mixture model, while others have a separate value for each component. This requires only minor changes to the M step of the EM algorithm. Instead of solving K separate optimization problems, we solve a single larger optimization problem of the form:

$$\text{argmax}_{\beta^1,\beta^2,\ldots,\beta^j} \left\{ \sum_j \sum_s \sum_i Z^j_{s,i} \log(L^j_{s,i}(\beta^j)) \right\}$$

subject to

$$\beta^1_z = \beta^2_z = \ldots = \beta^j_z \text{ for all shared } z.$$

Again, for g = logit, this is a weighted logistic regression problem that can be solved using a variety of standard algorithms.

5. EXPERIMENTS WITH GENERALIZED MODEL

In this section, we demonstrate the potential of the generalized mixture model by using it to learn a mixture of Additive Factor Models which models student performance on Duolingo listen challenges.

For these experiments, we use the same Duolingo dataset described in §3.1, but with all knowledge components included (i.e., every time student s completes a listen challenge, there is an example for each word w in the challenge, and the label for the example indicates whether the student included word w in their response). Each KC (i.e., each word) is considered a separate task. Note that although each listen challenge involves multiple KCs, we are using error attribution to create a dataset in which each example involves only a single KC. There is nothing about our methodology that requires this, but it mirrors the problem we wish to solve at Duolingo, and also allows for a cleaner comparison with Knowledge Tracing.

When splitting the data into training and test sets, we put each (student, KC) pair into one of the two groups uniformly at random. When fitting a mixture of Additive Factor Models, we use parameter sharing (see §4.2) for the student and KC indicator features, while allowing the times-seen feature to vary across components.

Figure 4 shows how performance on training and test data varies as a function of the number of components in the mixture model. The leftmost point ($K = 1$) corresponds to a regular Additive Factor Model, which can be fit by running a single logistic regression. Other points correspond to mixture models fit using the EM algorithm, in which each iteration entails solving a weighted logistic regression problem. As can be seen, using more than one component in the mixture model improves accuracy on both training and held-out test data.

Table 1 compares the performance of the Additive Factor

Model, the 3-component mixture of Additive Factor Models, Knowledge Tracing, and Performance Factors Analysis [16] on the same dataset. In this table, we present accuracy in terms of losses (log loss is -1 times log-likelihood, while AUC loss is one minus AUC), so lower values are better. As can be seen, the 3-component mixture gives the best performance of all the methods we considered in terms of both metrics, both on training and test data.

6. CONCLUSIONS

In this work we explored the use of mixture models to predict how students' error rates change as they learn. This led to order-of-magnitude improvements over Knowledge Tracing in terms of prediction accuracy on single-task datasets from Duolingo, as measured by the optimality gaps for both log-likelihood and AUC. Furthermore, examining the curves in the mixture model led us to uncover surprising facts about different groups of students.

We then generalized this mixture model to the multi-task setting, by learning a mixture of generalized linear models. This generalized mixture model offered state of the art performance on a large Duolingo dataset, outperforming Performance Factors Analysis, Additive Factor Models, and Knowledge Tracing on the same data.

There are several ways in which this work could be extended:

1. *Finding a good prior over learning curves.* In the single-task setting, we simply placed a Beta prior over each point on each learning curve. Though this worked well on the Duolingo dataset we considered (which contained around 15,000 data points), it may not give the best bias/variance tradeoff for smaller datasets. A natural way to constrain the algorithm would be to require error probability to be non-increasing as a function of trial number. Restricting to a particular family of curves such as exponentials or APEX functions [10], which generalize power laws and exponentials, may also be reasonable.

2. *Accounting for forgetting.* We have assumed that performance depends only on the trial number, and not on the amount of time elapsed since a particular knowledge component was last seen. For this reason, our model has no way to capture the benefit of spaced repetition [9] over massed practice, which is important for practice scheduling in the context of language learning [15].

3. *Feature exploration in the multi-task setting.* The generalized mixture model from §4 can be used with any set of features ϕ, but our experiments in §4 considered only a few possible choices. It would be interesting to explore other feature sets, and to see whether the features that work best in the usual regression setting ($K = 1$) are also best for larger K.

7. REFERENCES

[1] J. E. Beck. Using learning decomposition to analyze student fluency development. In *ITS 2006*, pages 21–28, 2006.

[2] J. E. Beck and K. Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146. Springer, 2007.

[3] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, pages 4–16, 1984.

[4] H. Cen, K. Koedinger, and B. Junker. Learning Factors Analysis – A general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.

[5] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.

[6] A. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In *User Modeling 2001*, pages 137–147. Springer, 2001.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[8] M. C. Desmarais and R. S. J. d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[9] H. Ebbinghaus. *Memory: A contribution to experimental psychology*. Dover, New York, 1885.

[10] A. Heathcote, S. Brown, and D. Mewhort. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7(2):185–207, 2000.

[11] B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283, 2011.

[12] P. McCullagh, J. A. Nelder, and P. McCullagh. *Generalized linear models*. Chapman and Hall London, 1989.

[13] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive skills and their acquisition*, pages 1–55. Erlbaum, Hillsdale, NJ, 1983.

[14] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[15] P. I. Pavlik and J. R. Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

[16] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis – A new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pages 531–538, 2009.

[17] Y. Xu and J. Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best? In *EDM*, pages 41–48, 2012.

Accounting for Slipping and Other False Negatives in Logistic Models of Student Learning

Christopher J. MacLellan
Human-Computer Interaction
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
cmaclell@cs.cmu.edu

Ran Liu
Human-Computer Interaction
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
ranliu@andrew.cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
koedinger@cmu.edu

ABSTRACT

Additive Factors Model (AFM) and Performance Factors Analysis (PFA) are two popular models of student learning that employ logistic regression to estimate parameters and predict performance. This is in contrast to Bayesian Knowledge Tracing (BKT) which uses a Hidden Markov Model formalism. While all three models tend to make similar predictions, they differ in their parameterization of student learning. One key difference is that BKT has parameters for the slipping rates of learned skills, whereas the logistic models do not. Thus, the logistic models assume that as students get more practice their probability of correctly answering monotonically converges to 100%, whereas BKT allows monotonic convergence to lower probabilities. In this paper, we present a novel modification of logistic regression that allows it to account for situations resulting in false negative student actions (e.g., slipping on known skills). We apply this new regression approach to create two new methods AFM+Slip and PFA+Slip and compare the performance of these new models to traditional AFM, PFA, and BKT. We find that across five datasets the new slipping models have the highest accuracy on 10-fold cross validation. We also find evidence that the slip parameters better enable the logistic models to fit steep learning rates, rather than better fitting the tail of learning curves as we expected. Lastly, we explore the use of high slip values as an indicator of skills that might benefit from skill label refinement. We find that after refining the skill model for one dataset using this approach the traditional model fit improved to be on par with the slip model.

Keywords

Cognitive Modeling, Statistical Models of Learning, Additive Factors Model, Performance Factors Analysis, Knowledge Tracing

1. INTRODUCTION

Statistical models of student learning make it possible for Intelligent Tutoring Systems [18] to be adaptive. These models estimate students' latent skill knowledge, so that tutors can use these estimates to intelligently select problems that give students more practice on skills that need it. Prior work has shown that even minor improvements in the predictive fit of latent knowledge models can result in less "wasted" student time, with more time on effective practice [22].

Two popular models of student learning are the Additive Factors Model (AFM) [4] and Performance Factors Analysis (PFA) [16]. Both are extensions of traditional Item Response Theory models [8]. While the two models differ in their parameterization of student learning, they both utilize logistic regression to estimate parameters and predict student performance. These models stand in contrast to other popular approaches like Bayesian Knowledge Tracing (BKT) [7], which uses Hidden Markov Modeling.

The BKT model is used both for "online" knowledge estimation within Intelligent Tutoring Systems (e.g., in Carnegie Learning's Cognitive tutor) to adaptively selecting practice items and for "offline" educational data modeling. The logistic models, on the other hand, have mainly been used in the context of offline data modeling. For example, DataShop, the largest open repository of educational data [12], uses AFM to fit student performance within existing datasets and to generate predicted learning curves. Data-driven cognitive task analyses, i.e., discovering and testing new mappings of tutor items to skills (or knowledge components), have used AFM as the core statistical model [17]. Novel knowledge component models can be discovered, evaluated in conjunction with AFM as a statistical model, validated on novel datasets [14], and used to guide tutor redesign efforts [13].

Despite the success of approaches like AFM, its lack of slip parameters has been emphasized as a key reason for favoring knowledge tracing over logistic models [10]. But knowledge tracing models tend to suffer from identifiability problems [1, 2]; e.g., the same performance data can be fit equally well by different parameters values, with different implications for system behavior. Furthermore, the actual effect of slip parameters on model predictions is complicated. The guess and slip parameters in BKT serve the dual purpose of modeling both noise, and the upper and lower bounds, in student performance. Without slip parameters, if a student gets an answer wrong, then BKT must assume that the student has not yet learned the skill. In contrast, the logistic models just model noise in the observations, so as long as the average student success rate converges to 100% then both models should perform similarly (assuming all other parameters are comparable across models). These approaches should only differ in situations where student performance converges to lower probabilities at higher opportunities; i.e., where false negatives such as slipping are actually occurring.

To investigate false negative phenomena, we augmented the logistic regression formalism to support slipping parameters. Using this new approach, which we call Bounded Logistic Regression, we produce two new student learning models: Additive Factors Model + Slip (AFM+Slip) and Performance Factors Analysis + Slip (PFA+Slip). These models are identical to their traditional counterparts but have additional parameters to model the false negative rates for each skill. We compare these models to their traditional counterparts and to BKT on five datasets across the domains of Geometry, Equation Solving, Writing, and Number Line Estimation. In all cases, the slip models have higher predictive accuracy (based on 10-fold cross validation) than their traditional counterparts.

We then move beyond comparing the predictive accuracies of the models to investigate how these parameters affect the predictions of the models and *why* these models are more accurate. Our analyses suggest that slipping parameters are not used to capture actual student "slipping" behavior (i.e., non-zero base rates for true student errors) but, rather, make the logistic models more flexible and allow better modeling of steeper learning rates while still predicting performance accurately at high opportunity counts (in the learning curve tail).

Lastly, we use AFM+Slip to perform data-driven refinement of the knowledge component (KC) model for a Geometry dataset. We identified a KC with a high false negative, or slip, rate and searched for ways to refine it. Using domain expertise, we refined the underlying KC model and showed that the traditional model (AFM) with the new KC model performed as well as the comparable slip model (AFM+Slip) did with the original KC model. This suggests that slip parameters allow the model to compensate for, and identify, an underspecified KC model.

2. STATISTICAL MODELS OF LEARNING

2.1 Logistic Models

The models in this class use logistic regression to estimate student and item parameters and to predict student performance. Thus, they model the probability that a student will get an step i correct using the following logistic function:

$$p_i = \frac{1}{1+e^{-z_i}}$$

where z_i is some linear function of student and item parameters for step i. The likelihood function for these models has been shown to be convex (i.e., no local maximums), so optimal parameter values can be efficiently computed and issues of identifiability only occur when there are limited amounts of data for each parameter. There are many possible logistic student learning models; in fact, most Item Response Theory models are in this class. For this paper, we will focus on two popular models in the educational data mining community: Additive Factors Model [4] and Performance Factors Analysis [16].

2.1.1 Additive Factors Model

This model utilizes individual parameters for each student's baseline ability level, each knowledge component's baseline difficulty, and the learning rate for each knowledge component (i.e., how much improvement occurs with each additional practice opportunity). The standard equation for this model is shown here:

$$z_i = \alpha_{student(i)} + \sum_{k \in KCs(i)} (\beta_k + \gamma_k \times opp(k,i))$$

where $\alpha_{student(i)}$ represents the prior knowledge of the student performing step i, the βs and γs represents the difficulty and learning rate of the KCs needed to solve step i, and $opp(k,i)$ represents the number of prior opportunities a student has had to practice skill k before step i. In the traditional formulation, the learning rates (γs) are bounded to be positive, so practicing KCs never decreases performance. To prevent the model from overfitting, the student parameters (αs) are typically L_2 regularized; i.e., they are given a normal prior with mean 0. Regularization decreases the model fit to the training data (i.e., the log-likelihood, AIC, and BIC) but improves the predictive accuracy on unseen data. Thus, when comparing regularized models to other approaches it should primarily be compared on measures that use held out data, such as cross validation.

2.1.2 Performance Factors Analysis

There are two key differences between this model and AFM. First, PFA does not have individual student parameters [16] (later variants have explored the addition of student parameters [6], but we base our current analysis on the original formulation). This usually substantially reduces the number of parameters of the model relative to AFM, particularly in datasets with a large number of unique students. Second, the model takes into account students' actual performance (not just opportunities completed) by splitting the learning rate for each skill into two learning rates: a rate for successful practice and a rate for unsuccessful practice. The standard equation based on these changes is the following:

$$z_i = \sum_{k \in KCs(i)} (\beta_k + \gamma_k success(i,k) + \rho_k failure(i,k))$$

where the βs represent the difficulty of the KCs, γs and ρs represent the learning rates for successful and unsuccessful practice on the KCs, $success(i,k)$ represents the number of successful applications of a skill k for the given student prior to step i, and $failure(i,k)$ represents the number of unsuccessful applications of a skill k for the given student prior to step i. Similar to AFM it is typical to restrict the learning rates (i.e., γs and ρs) to be positive [9]. One complication when comparing this model to other approaches using held out data (i.e., cross validation) is that the success and failure counts potentially contain additional information about the test data (i.e., performance on held out practice opportunities). Thus, we argue that comparing AFM to PFA using cross validation is usually not a fair comparison. Bearing this in mind, in the current analysis we were more interested in comparing AFM+Slip and PFA+Slip to their respective baseline models than to each other. To this end, we utilized cross validation as the primary measure of predictive accuracy for reasons previously discussed.

2.2 Bayesian Knowledge Tracing

There are many different models in the knowledge tracing family [10], but for this paper we focus on traditional 4-parameter BKT [7]. In contrast to the logistic approaches,

BKT utilizes a Hidden Markov Model to estimate latent parameters and predict student performance. This model has four parameters for each skill: the initial probability that the skill is known $p(L_0)$, the probability that the skill will transition from an unlearned to a learned state $p(T)$, the probability of an error given that the skill is learned $p(Slip)$, and the probability of a success when the skill is not learned $p(Guess)$. Unlike the logistic models, the estimation of these parameters can sometimes be difficult due to issues of identifiability [2] (e.g., there are many parameter values that yield the same likelihood) so these parameters are typically bounded to be within reasonable ranges; e.g., guess is typically bounded to be between 0 and 0.3 and slip is bounded to be between 0 and 0.1 [1]. Prior research has produced toolkits that can efficiently estimate these parameters using different approaches. For the comparisons in this paper we use the toolkit created by Yudelson et al. [23] and we use the gradient descent method.

One of the core differences between the logistic models and BKT is how they parameterize false negative student actions (i.e., slipping behavior). The logistic models do not have slip parameters and so they model student success as converging monotonically to 100% success (i.e., learning rates are bounded to be positive). In contrast, the BKT model explicitly models false negatives and allows monotonic convergence (under the typical assumption that the probability of forgetting is zero) to lower success rates. The slip parameters in BKT also serve the purpose of accounting for noise in student performance, and it is unclear whether these parameters account for true slipping behavior (i.e., non-zero base rate error) or just general noise in the student actions. Since the logistic models can already handle noise in the data, it remains to be seen what would happen if slip parameters were added to these models. That is the focus of this papers' investigation.

3. BOUNDED LOGISTIC REGRESSION

There is no trivial approach to incorporating explicit slip parameters into the logistic models; e.g., the prediction probability cannot be bounded by an additional linear term to the logistic function. In order to add these parameters we modified the underlying logistic model to have the following form:

$$p_i = \frac{1}{1+e^{-s_i}} \times \frac{1}{1+e^{-z_i}}$$

where z_i is the same as that used in standard logistic regression and s_i is a linear function of the parameters that impose an upper bound on the success probability for the step i. For modeling a slip rate for each skill we use the following equation:

$$s_i = \tau + \sum_{k \in KCs(i)} \delta_k$$

where τ is the parameter corresponding to the average slip rate across all items and students and δ_k is the change in the average slip rate for each skill k. We also apply an L_2 regularization to the δ parameters to prevent overfitting. To fit the parameters we used the sequential quadratic programming package in Octave, which uses an approach similar to Newton-Raphson but properly accounts for parameter constraints (e.g., positive learning rates). For details on parameter estimation see Appendix A.

This formulation is a generalization of Item Response Theory approaches that model item slip (e.g., [21]). In particular, it supports slipping with multiple KC labels per an item by using a logistic function to map the sum of slip parameters to a value between 0 and 1. For items with a single KC label, the $\frac{1}{1+e^{-s_i}}$ term reduces to the slip probability for that KC. For multi-KC items, this term models slipping as the linear combination of the individual KC slipping parameters in logit space. This approach mirrors that taken by AFM and PFA for modeling KC difficulty and learning rates in situations with multiple KC labels. In these situations, prior work has shown that the logit approach gives a good approximations of both conjunctive and disjunctive KC behavior [4].

During early model exploration we used Markov Chain Monte Carlo methods to compare this formulation with a more complex formulation that had parameters for both guessing and slipping. Our preliminary results showed that AFM with slip parameters outperformed the guess-and-slip variation for the 'Geometry Area (1996-97)' [11] and the 'Self Explanation sch_a3329ee9 Winter 2008 (CL)' [3] datasets (accessed via DataShop [12]) in terms of deviance information criterion (a generalization of AIC for sampled data). Further analysis showed that there was little data to estimate the guessing portion of the logistic curve. This is because the average student error rate in these datasets starts off at less than 50% and only gets lower with practice. This is typical of many of the available tutor datasets, so for our Bounded Logistic Regression approach we decided it would be sufficient to model the slipping parameters.

4. EVALUATION

4.1 Method

We used bounded logistic regression to add slip parameters to AFM and PFA, thus creating two new student learning models: AFM + Slip and PFA + Slip. We were interested in how these approaches compared with their traditional counterparts and to Bayesian Knowledge Tracing, which parameterizes guess and slip. Furthermore, we were interested in how these different approaches compared across different datasets spanning distinct domains. To perform this evaluation we fit each of the five models to five datasets we downloaded from DataShop [12]: Geometry Area (1996-97) [11], Self Explanation sch_a3329ee9 Winter 2008 (CL)[3], IWT Self-Explanation Study 1 (Spring 2009) (tutors only) [19], IWT Self-Explanation Study 2 (Fall 2009) (tutors only) [20], and Digital Games for Improving Number Sense - Study 1 [15]. These datasets cover the domains of geometry, equation solving, writing, and number line estimation. We selected these datasets because they have undergone extensive KC model refinement, including both manually created models by domain experts and automatically-refined models using Learning Factors Analysis [5]. For all datasets we used the best fitting KC model, based on unstratified cross validation.

In addition to comparing the different statistical models' predictive accuracies, we were interested in understanding

Table 1: In all five datasets the slip models outperform their non-slip counterparts in terms of log-likelihood and cross validation. In four out of the five datasets, the PFA+Slip model outperforms the AFM+Slip model in terms of log-likelihood and cross validation performance. In this table "Par." represents the number of parameters in the model and the CV RMSE values are the averages of 10 runs of 10-fold un-stratified cross validation.

Dataset	Model	Par.	LL	AIC	BIC	CV RMSE
Geometry						
	AFM	95	-2399.7	4989.4	5610.5	0.396
	AFM+Slip	114	-2377.0	4982.0	5727.3	0.395
	PFA	54	-2374.9	4857.8	**5210.8**	0.389
	PFA+Slip	73	**-2298.3**	**4742.6**	5219.8	**0.383**
	BKT	72	-2460.8	5065.7	5536.5	0.396
Equation Solving						
	AFM	106	3011.6	6235.2	6953.9	0.390
	AFM+Slip	125	**-2992.5**	**6235.0**	7082.54	**0.388**
	PFA	48	-3205.2	6506.4	6831.8	0.400
	PFA+Slip	67	-3088.9	6311.8	**6766.0**	0.392
	BKT	72	-3202.7	6549.5	7037.7	0.426
Writing 1						
	AFM	169	-3214.6	6767.2	7916.1	0.406
	AFM+Slip	196	-3214.6	6821.2	8153.6	0.406
	PFA	72	-3212.0	6568.0	**7057.4**	0.401
	PFA+Slip	99	**-3158.0**	**6514.0**	7187.0	**0.398**
	BKT	104	-3480.2	7168.5	7875.6	0.419
Writing 2						
	AFM	129	-2976.4	6210.8	7096.6	0.375
	AFM+Slip	145	-2962.8	6215.6	7211.3	0.373
	PFA	45	-2994.7	6079.4	**6388.4**	0.373
	PFA+Slip	61	**2965.7**	**6053.4**	6472.2	**0.371**
	BKT	60	-3177.1	6474.2	6886.2	0.384
Number Line						
	AFM	93	-2352.7	4891.4	5484.0	0.433
	AFM+Slip	115	-2356.3	4942.6	5675.4	0.432
	PFA	62	-2337.5	**4799.0**	**5194.1**	0.430
	PFA+Slip	84	**-2318.9**	4805.8	5341.1	**0.428**
	BKT	84	-2548.7	5265.4	5800.7	0.451

and interpreting the situations in which slip parameters improve model fit. Prior to analysis we hypothesized that slipping parameters might have three potential effects on the model fit: (1) enabling the model to capture true student slipping behavior; i.e., KCs that have a non-zero base-rate error, (2) enabling the model to fit steeper initial learning rates while still making correct predictions at higher opportunity counts, and (3) enabling the model to compensate for an underspecified knowledge component model. We focused in on one dataset, Geometry Area (1996-97), to explore these possibilities. Within this dataset we conducted a residual analysis to explore possibilities (1) and (2). We then refined the geometry KC model for a specific KC that the slip model identified as having a high false negative rate (i.e., slip value) to explore possibility (3). For brevity we only show the results of AFM and AFM+Slip in these analyses, but similar trends hold for PFA and PFA+Slip.

4.2 Results

4.2.1 Model Fits for Five Datasets

We fit each of the five models to the five datasets. Table 1 shows the resulting model fit statistics and the number of parameters used in each model. AFM has 1 parameter per student and 2 parameters per skill, PFA has 3 parameters for each skill, and BKT has 4 parameters for each skill. The slip variations have an additional parameter for each skill, plus a parameter for the average slip rate. When using the PFA models in practice many of the KCs never had any unsuccessful practice (i.e., their failure count was always 0). In these situations we removed the parameters for the failure learning rates because they have no effect on the model behavior. Thus, in some situations, the number of parameters in each model might differ from the general trends. All of the cross validation results are the average of 10 runs of 10-fold unstratified cross validation, where the cross validated RMSE was computed using the predicted probability of a correct response (rather than discrete correct/incorrect predictions).

All of the slip models have better log-likelihood and cross validation performance than their respective baseline models (AFM and PFM). Furthermore, in four out of the five datasets, PFA+Slip has better cross validation performance than AFM+Slip, even though it does not have individual student parameters. Finally, all of the logistic models outperformed traditional four-parameter BKT. Based on prior work [16] we expected this last result, but we included BKT as a comparison model that supports slipping. In particular,

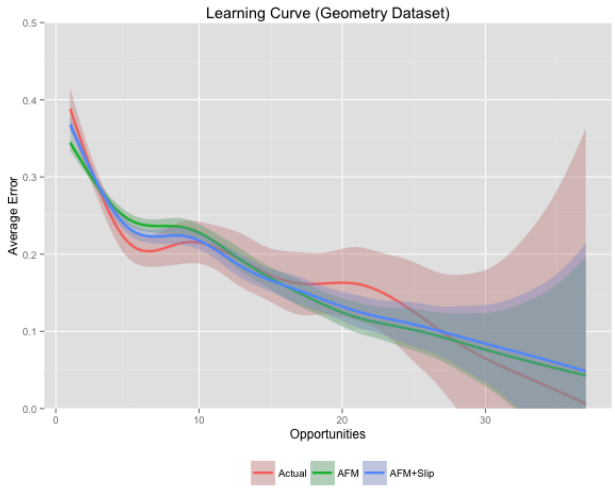


Figure 1: The AFM+Slip model better fits the steeper learning rate of the Geometry dataset than the AFM model, but both models fit the tail of the learning curve reasonably well and the actual student error appears to be converging to 0%. The shaded regions denote the 95% confidence intervals for the respective values.

Figure 2: The 95% confidence intervals (shaded regions) for the residuals of the AFM model do not include zero for lower opportunity counts, the model first overpredicts and then underpredicts success. In contrast the 95% confidence intervals for residuals of the AFM+Slip model always include zero indicating a better model fit.

Figure 3 shows an example of how the AFM+Slip model fits the data more like the BKT model than the AFM model for a KC with a high slip rate.

4.2.2 *Residual Analysis*

To investigate how the predictions of the slip models differ from that of the traditional models we analyzed the residuals for the AFM and AFM+Slip models on the Geometry dataset. Figure 1 shows the actual and predicted error rates for the two models on this dataset and Figure 2 shows the model residuals plotted by opportunity count. Investigating patterns in residual error across opportunity counts is a useful way of assessing systematic discrepancies between a given model's predicted learning curves and students' actual learning curves.

Although both models fit the data reasonably well, the slip model better models the steepness at the beginning of the learning curve. At low opportunity counts, AFM without slip typically predicts a substantially flatter learning curve compared to the actual data. The residual plot mirrors this finding; the 95% confidence interval for the AFM residuals does not include zero for earlier opportunities and the model flips from over-predicting success to under-predicting it. The AFM+Slip model, in contrast, better models the initial steepness of the learning curve. The 95% confidence interval for the AFM+Slip model residuals always includes zero. Finally, we see no evidence of actual slipping behavior in the tail of the learning curve: the 95% confidence intervals for residuals in both models include zero for higher opportunity counts. If true student slipping were occurring, we would expect the traditional AFM model to overpredict success in the tail, but we do not observe this.

4.2.3 *KC Refinement Based on False Negatives*

In order to explore the hypothesis that a high false negative, or slip, rate on a skill is indicative of a underspecified knowledge component model, we analyzed a KC on which the slip parameter was high and on which AFM and AFM+Slip differed substantially in their predictions. One KC, "geometry*compose-by-multiplication," fit this criteria. Figure 3 shows the learning curve with model predictions for this KC. AFM+Slip makes predictions that are nearly identical to BKT and seems to better fit the actual student learning curve. Upon further investigation, we found that many of the items labeled with this skill were on the same problems. Within these problems, we noticed that the later problem steps (items) might actually have been solved by applying the "arithmetic" skill to the result of an earlier application of the "compose-by-multiplication" skill. We generated a new knowledge component model to reflect these findings and re-fit the model using AFM. The predictions of this new model (AFM-New-KC) are also shown in Figure 3. For the AFM-New-KC plot, we plotted the observations with the opportunity counts from the original KC model (x-axis) but with predicted errors from the new KC model (y-axis). This was necessary for the purposes of comparison to the original KC model predictions. Once the knowledge component model was refined based on the insights provided by fitting AFM+Slip, standard AFM improved. Furthermore, based on this change the overall AFM model fit improved to be on par with AFM+Slip in terms of log-likelihood, AIC, and cross validation (LL = -2378.8, AIC = 4947.6, BIC = 5568.6, and CV RMSE = 0.395).

5. DISCUSSION

Our model fit results show that the slip models have better predictive accuracy (i.e., cross validation performance) and

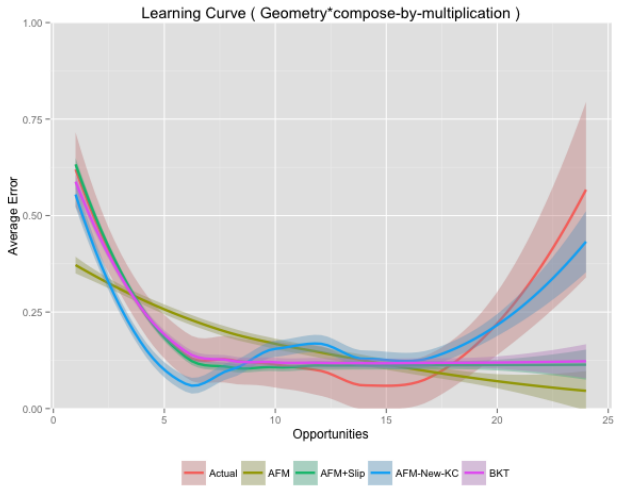


Figure 3: AFM+Slip looks much more like BKT for this KC and seems to model the data better (the overlapping purple and green lines). We took the high false negative rate (i.e., the sharp floor in the predicted error at approx. 11%) as an indicator that the KC model might benefit from refinement. Refitting the regular AFM model with a refined KC model (AFM-new-KC) shows a better fit to the actual data. Shaded regions denote the 95% confidence intervals for the respective values.

log-likelihood fits than their traditional counterparts across all five datasets. Furthermore, the AIC scores generally mirror this finding. These results suggest that the addition of the slip parameters to the logistic model formalism results in an improved model fit and an increased ability to predict behavior on unseen data.

In four of the five datasets, PFA + Slip best fit the data in terms of both log-likelihood and cross validation. In one sense, its superior cross-validation performance is surprising because the PFA models (as implemented here) have no student intercept parameters. However, they have an advantage for the cross validation statistic because they get success and failure counts that include information about performance on held out data, essentially giving these models an advantage over the other models. The better log-likelihood (and often AIC) scores are indicative of a better ability to fit the data that doesn't suffer from this discrepancy. However, PFA models have an advantage over AFM for this statistic because AFM uses regularization, which intentionally worsens the fit of the model to the data in an effort to improve predictive accuracy. To test if regularizing student parameters was causing PFA and PFA + Slip to outperform AFM and AFM + Slip we refit the AFM models to the Geometry dataset with student parameter regularization disabled and found that, at least for the Geometry dataset, the PFA models still outperforms the AFM models in terms of log-likelihood, AIC, BIC, and CV RMSE. These findings suggest that the PFA models better fits the data than the AFM models, but more work is needed to explore how best to compare these two approaches and to determine when one approach is preferable to another.

Lastly, the logistic models consistently outperform traditional four-parameter BKT. This is somewhat unsurprising because BKT does not have individual student parameters or separate learning rates for success and failure. However, we still included traditional BKT as a baseline model that is widely used and has explicit parameters for guess and slip. In particular, Figure 3 shows that for a KCs with high slip rate the AFM+Slip model performs more like BKT than AFM, suggesting that the new model is able to fit slipping and other false negative student behavior.

Given the finding that the slip models have better predictive accuracy and log-likelihood fits than their traditional counterparts, we investigated how the addition of slip parameters changed the model predictions. Residual analyses on the Geometry dataset showed that both AFM and AFM+Slip had similar fits to the data, but AFM+Slip better fit the initial steepness of the learning curve while maintaining a good fit in the tail. This intuition is confirmed in the residual by opportunity plot, which shows that the 95% confidence intervals for the residuals from AFM exclude zero at low opportunity counts, first overpredicting success and then underpredicting it. In contrast, the 95% confidence interval for the residuals from AFM+Slip include zero at these same low opportunity counts. This evidence supports the hypothesis that adding slip parameters enables the model to better accommodate steeper learning rates. In contrast, we find no evidence to support the hypothesis that adding slipping parameters enables the model to better fit non-zero base rate error; i.e., true student slipping. If this were the case, then we would expect AFM to overpredict success in the tail (i.e., for the residuals to be non-zero at higher opportunity counts), but we found no evidence that this occurred.

Finally, we demonstrated that high false negative, or slip, rates can serve as detectors of KCs that might benefit from further refinement. We identified a KC in the Geometry dataset that had a high slip rate and that differed from the traditional model: the "geometry*compose-by-multiplication" KC. We found that this KC could be further refined and showed that AFM with the refined KC model performed on par with AFM+Slip in terms of log-likelihood and cross validation. This suggests that adding slip parameters to a model can enable it to compensate for an underspecified KC model but, more importantly, can help identify these poorly specified KCs. The newly discovered KC model better fit the student data than the previous best model, which was the result of years of hand and automated KC model refinement.

6. CONCLUSIONS

Logistic models of learning, such as AFM and PFA, are popular approaches for modeling educational data. However, unlike models in the knowledge tracing family, they do not have the ability to explicitly model guessing and slipping rates on KCs. In this work we augmented traditional logistic regression to support slipping rates using an approach that we call Bounded Logistic Regression. We then used this approach to create two new student models: AFM + Slip and PFA + Slip. We then compared the performance of these new models in relation to their traditional counterparts. Furthermore, for AFM we explored how the addi-

tion of slip parameters changed the predictions made by the model. We explored three possibilities: (1) they might enable the model to capture true student slipping behavior (i.e, non-zero base-rate error), (2) they might enable the model to accommodate steeper learning rates while still effectively predicting performance at higher opportunity counts, and (3) they might enable the model to compensate for an underspecified knowledge component model.

To explore the first two possibilities, we conducted a residual analysis and found that the slip parameters appear to help the model fit steeper learning rates, rather than improving model fit in the tail. To explore the third possibility, we used a high false negative, or slip, rate as an indicator of where the given KC model might benefit from refinement. We found that after refining a KC model using this approach AFM performance (e.g., CV, LL, AIC) improved to be on par with AFM-Slip. This suggests that the slip parameters enable the model to compensate for underspecified KC models and that high slip values can be used to identify KCs that might benefit from further KC label refinement.

7. LIMITATIONS AND FUTURE WORK

One key limitation of the current work is that we did not explore issues of identifiability in the Bounded Logistic Regression model. In particular, we have not yet demonstrated that the log-likelihood for models using this formalism are convex. In the current formulation we only model slip parameters (not guess parameters), so we expect identifiability to be less of an issues. In line with this intuition we found that the current approach returned reasonable parameter values and consistently improved model fit across the five data sets we explored. However, we recognize that the model would benefit from a more rigorous analysis of the quality of estimated parameters and acknowledge that this would be an important direction for future work.

Finally, the current work focuses on comparing the slip models to their traditional counterparts, but future work might explore how different models (e.g., AFM+Slip, PFA+Slip, and BKT) compare to one another. In the current work we purposefully avoided making conclusions about how these models compare because there is some ambiguity in how different approaches are evaluated. For example, Yudelson's Bayesian Knowledge Tracing toolkit [23] performs incremental prediction during cross validation (i.e., predicting student performance on a step and then "showing" the model the actual performance before moving on to the next step). While this approach aligns well with the actual use of the BKT model it gives an unfair advantage when comparing it to cross validated AFM, which gets no information about test data when making predictions. A similar complication exists for PFA, which gets information about the performance of unseen steps from the success and failure counts. A more equivalent comparison would be to perform an incremental prediction using AFM and PFA, but this was beyond the scope of the current paper and represents an open area for future work.

8. ACKNOWLEDGEMENTS

We thank Erik Harpstead, Michael Yudelson, and Rony Patel for their thoughts and comments when developing this work. This work was supported in part by the Department of Education (#R305B090023 and #R305B110003) and by the National Science Foundation (#SBE-0836012). Finally, we thank Carnegie Learning and all other data providers for making their data available on DataShop.

9. REFERENCES

[1] R. S. J. d. Baker, A. T. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In B. P. Woolf, E. Aimeur, R. Nkambou, and L. S, editors, *ITS '08*, pages 406–415, 2008.

[2] J. E. Beck and K.-M. Chang. Identifiability: A Fundamental Problem of Student Modeling. In C. Conati, K. McCoy, and G. Paliouras, editors, *UM '07*, pages 137–146, 2007.

[3] J. Booth and S. Ritter. Self Explanation sch_a3329ee9 Winter 2008 (CL). `pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=293`.

[4] H. Cen. *Generalized Learning Factors Analysis: Improving cognitive Models with Machine Learning.* PhD thesis, Carnegie Mellon University, 2009.

[5] H. Cen, K. R. Koedinger, and B. Junker. Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. In M. Ikeda, K. Ashlay, and T.-W. Chan, editors, *ITS '06*, pages 164–175, 2006.

[6] M. Chi, K. R. Koedinger, G. Gordon, P. Jordan, and K. Vanlehn. Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, and J. Stamper, editors, *EDM '11*, pages 61–70, 2011.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278, 1995.

[8] K. L. Draney, P. Pirolli, and M. Wilson. A measurement model for a complex cognitive skill. In P. N, S. Chipman, and R. Brennan, editors, *Cognitively diagnostic assessment*, pages 103–125. Lawrence Erlbaum Associates Inc., 1995.

[9] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In V. Aleven, J. Kay, and J. Mostow, editors, *ITS '10*, pages 35–44, 2010.

[10] G.-B. Jose, H. Yun, and P. Brusilovsky. General Features in Knowledge Tracing: Applications to Multiple Subskills,Temporal Item Response Theory, and Expert Knowledge. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren, editors, *EDM '14*, pages 84–91, 2014.

[11] K. Koedinger. Geometry Area 1996-1997. `pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76`.

[12] K. R. Koedinger, R. S. J. d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A Data Repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, and R. S. J. d. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.

[13] K. R. Koedinger, J. Stamper, E. McLaughlin, and

T. Nixon. Using Data-Driven Discovery of Better Student Models to Improve Student Learning. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *AIED '13*, pages 421–430, 2013.

[14] R. Liu, K. R. Koedinger, and E. A. McLaughlin. Interpreting Model Discovery and Testing Generalization to a New Dataset. In *EDM '14*, pages 107–113, 2014.

[15] D. Lomas. Digital Games for Improving Number Sense - Study 1. `pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=445`.

[16] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance Factors Analysis –A New Alternative to Knowledge Tracing. In V. Dimitrova and R. Mizoguchi, editors, *AIED '09*, pages 531–538, 2009.

[17] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using Data. In J. Kay, S. Bull, and G. Biswas, editors, *AIED '11*, pages 353–360, 2011.

[18] K. Vanlehn. The Behavior of Tutoring Systems. *IJAIED*, 16(3):227–265, 2006.

[19] R. Wylie. IWT Self-Explanation Study 1 (Spring 2009) (tutors only). `pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=313`.

[20] R. Wylie. IWT Self-Explanation Study 2 (Spring 2009) (tutors only). `pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=372`.

[21] Y. C. Yen, R. G. Ho, W. W. Laio, L. J. Chen, and C. C. Kuo. An Empirical Evaluation of the Slip Correction in the Four Parameter Logistic Models With Computerized Adaptive Testing. *APM*, 36(2):75–87, 2012.

[22] M. V. Yudelson and K. R. Koedinger. Estimating the benefits of student model improvements on a substantive scale. In S. K. D'Mello, R. A. Calvo, and A. Olney, editors, *EDM '13*, 2013.

[23] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian Knowledge Tracing Models. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *AIED '13*, pages 171–180, 2013.

APPENDIX

A. PARAMETER ESTIMATION

Similar to standard logistic regression we assume the data follows a binomial distribution. Thus, the likelihood and log-likelihood are as follows:

$$\begin{aligned} Likelihood(data) &= \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)} \\ ll(data) &= \sum_{i=1}^{n} y_i log(p_i) + (1-y_i) log(1-p_i) \end{aligned}$$

where y_i is 0 or 1 depending on if the given step i was correct. As mentioned earlier, p_i is defined as:

$$p_i = \frac{1}{1+e^{-s_i}} \times \frac{1}{1+e^{-z_i}}$$

where s_i is the linear combination of the slip parameters and z_i is the linear combination of the student and item parameters.

To estimate the parameters values for bounded logistic regression, we maximize the conditional maximum likelihood of the data using sequential quadratic programming (specifically the sqp package in Octave). This approach reduces to applying the Newton-Raphson method, but properly accounts for situations when the parameter values are constrained, such as the positive bound for the learning rates in AFM and PFA. To apply this method, we needed to compute the gradient and hessian for the likelihood of the data given the model.

To compute the gradient we took the derivative with respect to the student and item parameters (w's) and slip parameters (sp's). For the student and item parameters the gradient is the following:

$$\frac{dll}{dw_a} = \sum_{i=1}^{n} \frac{x_{ia}}{1+e^{z_i}} \frac{(y_i - p_i)}{(1-p_i)}$$

where x_{ia} is the value of the student or item feature that is being weighted by parameter w_a for step i.

Similarly, for the slip parameters the gradient is the following:

$$\frac{dll}{dsp_a} = \sum_{i=1}^{n} \frac{q_{ia}}{1+e^{s_i}} \frac{(y_i - p_i)}{(1-p_i)}$$

where q_{ia} is the value of the slip feature (in AFM and PFA these are the 0 or 1 entries from the Q-matrix) that is being weighted by parameter sp_a for step i.

Given these gradients we have a hessian matrix with values for the interactions of the ws with each other, the ws with the sps, and the sps with each other. These values are defined as the following:

$$\begin{aligned} \frac{d^2ll}{dw_a dw_b} &= \sum_{i=1}^{n} \frac{x_{ia}x_{ib}}{(1+e^{z_i})^2(1-p_i)^2}[p_i(y_i-1) \\ &\quad + e^{z_i}(p_i - y_i)(1-p_i)] \\ \frac{d^2ll}{dsp_a dsp_b} &= \sum_{i=1}^{n} \frac{q_{ia}q_{ib}}{(1+e^{s_i})^2(1-p_i)^2}[p_i(y_i-1) \\ &\quad + e^{s_i}(p_i - y_i)(1-p_i)] \\ \frac{d^2ll}{dw_a dsp_b} &= \sum_{i=1}^{n} \frac{x_{ia}}{1+e^{z_i}} \left[\frac{(p_i-1)+(y_i-p_i)}{(1-p_i)^2}\right] \end{aligned}$$

Finally, in our formulation we applied an L_2 regularization to all of the parameter values (i.e., a normal prior with mean 0), where the λ parameter of the regularization could be set individually for each model parameter. For the AFM models we set λ to 1 for the student parameters. For all of the slip models we λ to 1 for the KC slip parameters (i.e., δs). For all other parameters we turned regularization off ($\lambda = 0$).

Learning Environments and Inquiry Behaviors in Science Inquiry Learning: How their Interplay Affects the Development of Conceptual Understanding in Physics

Engin Bumbacher* buben@stanford.edu
Shima Salehi* salehi@stanford.edu
Miriam Wierzchula miriamw1989@gmail.com
Paulo Blikstein* paulob@stanford.edu

* Stanford University, CERAS 102, 520 Galvez Mall, Stanford, CA, 94305

ABSTRACT

Studies comparing virtual and physical manipulative environments (VME and PME) in inquiry-based science learning have mostly focused on students' learning outcomes but not on the actual processes they engage in during the learning activities. In this paper, we examined experimentation strategies in an inquiry activity and their relation to conceptual learning outcomes. We assigned college students to either use VME or PME for a goal-directed physics inquiry task on mass-spring systems. Our analysis showed that the best predictors of learning outcomes were experimental manipulations that followed a control of variable (CV) strategy, with a delay between manipulations ("systematic inquiry"). Cluster analysis of the prevalence of these manipulations per participant revealed two distinct clusters of participants, systematic inquiry or not. The systematic inquiry cluster had significantly higher learning outcomes than the less systematic one. Furthermore, the majority of the participants using the PME belonged to the more systematic cluster, while most of the participants using the VME fell into the non-systematic cluster, likely because of the specific affordances of the real and virtual equipment they were using. However, beyond this impact on inquiry process, condition had little effect. In light of these results, we argue that investigating processes displayed during learning activities, in addition to outcomes, enables us to properly evaluate the strengths and weaknesses of different learning environments for inquiry-based learning.

Keywords

Science Discovery Learning, Computer Simulations, Real Laboratories, Inquiry Learning, Cluster Analysis, Virtual and Physical Science Laboratories

1. Introduction

Over the past decades, the science teaching community has adopted the view that "students cannot fully understand scientific and engineering ideas without engaging in the practices of inquiry and the discourses by which such ideas are developed and refined" (NRC, 2012, p.218). Inquiry-based instruction requires students to model the practices of scientific inquiry to actively develop their conceptual understanding [1,2]. While physical laboratories were the traditional environments for such inquiry-based learning, there is accumulating evidence that virtual laboratories are similarly well suited to meet the goals of science investigation [3,4]. In particular, they are considered to be at least equally conducive to active manipulations for experimentation [2,3], which is seen as the crucial aspect of inquiry learning [5,6,7].

A major limitation of the research comparing physical and virtual manipulative environments (PME and VME) for science learning was the predominant focus on the learning *outcomes* rather than the learning *processes* when students engage in inquiry activities. This has not changed with recent work that shifted from treating the environments as two competing entities to examining how to best combine them for increased learning benefits [4]. We argue that research on how learners engage with these manipulative environments could provide a more comprehensive understanding of how the interaction of a learner with an environment impacts the learners' construction of knowledge, and in turn what design features of these environments foster desired manipulative behaviors in the context of science inquiry learning.

The present study lies at the intersection of research on learning environments and research on inquiry behaviors in order to study the characteristics of productive experimentation strategies in open-ended science investigation tasks, and how such strategy use might be influenced by the different affordances of the learning environments. For this purpose we encoded the actual experiments students ran, which allows us to basically replay their processes. This allows us to explore customized operationalizations of inquiry behaviors of interest. This approach integrates data-driven methods with relevant theoretical concepts. As a result, we found a robust characterization of experimentation strategies that meaningfully predicts learning outcomes, and show how participants' strategy use differs between the learning environments. This study is part of a larger research project with the goal of developing automated detectors of systematic inquiry in open-ended science investigation activities for formative assessment and for the design of productive learning environments.

2. Inquiry Behaviors

2.1. Control of Variable Strategy

Scientific learning through self-directed inquiry activities depends on the actual inquiry behaviors employed [8,9]. In particular, adequate experimentation strategies are required that result in interpretable observations, i.e. evidence that facilitates drawing valid inferences. Research has particularly focused on the abilities to systematically combine variables and to design unconfounded experiments, i.e. experiments that modify variables such that competing hypothesis can be ruled out. The design of unconfounded experiments requires the ability to employ the *control of variables strategy* (CVS), that is, to create experiments with a single contrast between experimental conditions [10]. This is in contrast to inadequate strategies such as changing multiple variables at the same time, which hampers valid inferences and subsequent knowledge [11].

Previous research has examined a host of individual and contextual factors of strategy use [8]. However, only a very small number of studies have explicitly examined the impact of affordances of learning environments on strategy use in experimentation activities [2,12]. While Triona & Klahr [2] focused on the impact of physicality of manipulatives alone on

learning outcomes, Renken & Nunez [12] had students engage in an inquiry activity on pendulum motion using either a PME or a VME that differed in both ease of manipulation and freedom of choice: while the PME provided participants with only three different levels for either pendulum length or mass, the VME allowed participants to modify the variables smoothly by means of continuous valued sliders. Even if there was no difference in conceptual understanding between the VME and PME conditions, participants using the computer simulation ran more trials and were less likely to control variables. Renken and Nunez [12] argued that the additional flexibility and breadth of choice in experimentation in VME was detrimental to participants' use of adequate experimentation strategy.

While this study suggests that indeed strategy use in inquiry-based learning activities is influenced by affordances of the learning environments, it is difficult to generalize these results to less structured and scaffolded inquiry activities.

2.2. Operationalization of Inquiry Strategies

As most studies cited mainly focused on CV strategy, they used highly structured tasks where either variables were dichotomous, or there was only one outcome variable, or the activity was restricted. In order to develop a more nuanced characterization of inquiry strategies, we need more complex inquiry tasks. Data mining techniques employed in such contexts have been successful at discovering groups of similar users [13,14,15]. Most of these data-mined systems are based on the user interaction logs [16]. While they achieve good predictive power, such machine-learned detectors of interaction behaviors often come at the cost of interpretability [17]. However, it is crucial to develop data-mined models of inquiry strategies that are interpretable in order to advance our understanding of learning processes through inquiry activities. We apply a different approach, where we do not use labelled action logs but code the actual experiment configurations of each participant. Based on video data, we extract each configuration a participant tried and feed it into a database of experiments of all participants. This allows us to quickly extract and explore relevant variables of inquiry, such as the number of spring-only or mass-only changes, the number of unique configurations, repetitions, etc. That way, we can integrate relevant theoretical concepts into the operationalization of inquiry behaviors.

In the context of this study, we focused on experimentation strategies only. We collected data on the number of experiment trials, the experiment configurations, and the time between manipulations, and coded the type of manipulation per experiment. Particular focus is given to "*control of variable" manipulations, "deliberate" manipulations, and "deliberate control" of variable manipulations*. Deliberate manipulations (DM) are manipulations into which a participant has put some thought, as measured by *dwell time* between two consecutive manipulations. We assume that participants who are cognitively engaged – reflecting on evidence from a preceding manipulation, trying to make sense of it in the context of previous observations, or taking notes or planning the next manipulation(s) – will spend more time before executing the next change than those who are cognitively less engaged.

For this reason, we include the third category of manipulations that lies at the intersection of the prior two categories, *deliberate control of variable manipulations* (DCVM). As prior research on experimentation strategies in inquiry-based activities characterized them as solely CVS or not, activities were designed such that controlling variables in an experiment had to be a deliberate choice of participants [19,20,21]. However, in less structured, open-ended inquiry like those used in this study, it is possible in some cases to manipulate variables according to a CV strategy without the deliberate intention to do so. For example in the computer simulation for our mass & spring activity, one could change the value of the spring constant continuously by means of a slider, without having to interrupt an ongoing experiment. Inherently, this corresponds to a control of variable manipulation (CVM) but not necessarily to a *deliberate* control of variable manipulation (DCVM).

3. Present Study

The study reported here was part of a larger study examining participants' inquiry behaviors in different scientific domains using either PME or VME as learning environments. Participants engaged in two activities; the first one was on mass and spring oscillation (see Figure 1), and the second one on basic electric circuits. The current paper presents analysis of the first inquiry activity. During the first activity, participants were either asked to simply think-aloud while engaging in the inquiry or were trained to implement the Predict-Observe-Explain framework (POE) [18]. The training session of the POE framework was highly structured and guided: During the entire activity, before each intended manipulation, participants were asked to predict its result, then observe the actual results of the manipulation, and finally explain their observation in light of the initial prediction. On the other hand, the think-aloud group did not receive any scaffolds or guidance by the experimenter. Therefore, for the purposes of this paper, we report only data for the participants in the think-aloud condition, as the difference in guidance might have altered the nature of the activity, and masked the effect of medium on inquiry processes of the participants.

The main research questions that guided the present study were:

- How can we operationalize inquiry strategies in less well-structured and more complex activities?
- What inquiry strategies are related to better learning outcomes?
- How does strategy use differ between participants using either the physical or the simulation environment?

3.1. Sample

For Mass and spring activity in think-aloud condition, we had 36 community college students (24 female, 12 male, average age=20.5, SD=3.6).

3.2. Design

The study reported here is a between subject design with two levels. We randomly assigned participants to use either *physical* (PHY) or *computer simulation* (SIM) to engage in an inquiry-based activity on mass and spring oscillation (n_{PHY}=18, n_{SIM}=18). The task was to discover how the mass and the spring constant affect both the amplitude and the frequency of oscillation of a mass-spring system. We administrated a conceptual test before and after the activity. The post-test scores were the dependent measures of the experiment, while the pre-test scores were used as covariates in the corresponding statistical analyses. The relevant

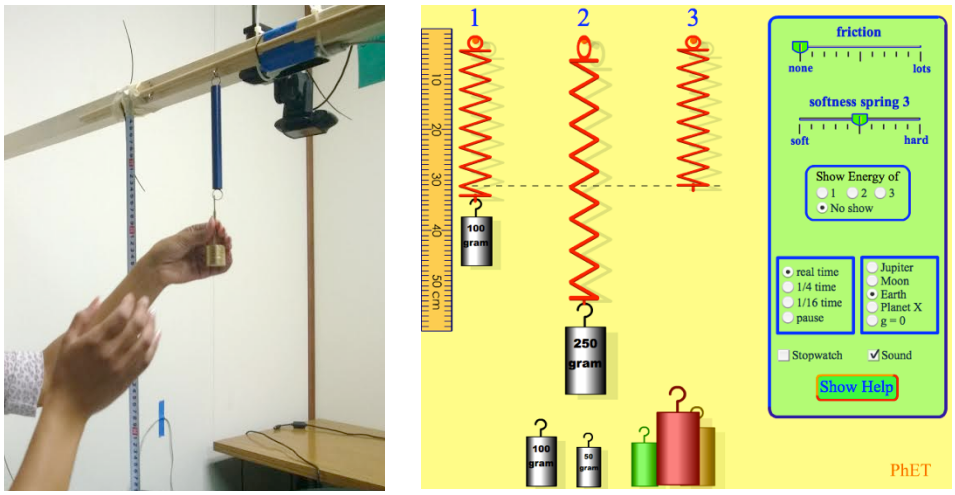


Figure 1. Experimental Setup: Left: Physical toolkit in action: The first hook is just next to the measure tape. Right: Computer Simulation: Participants were only allowed to change the "softness spring 3".

behavioral measures were treated as independent within-subject variables since they were expected to predict learning outcomes.

3.3. Materials

3.3.1. Learning Environment

Physical Learning Environment. The physical toolkit consisted of the PASCO[1] Demonstration Spring Set and Mass and Hanger Set. There are four pairs of springs, each with a spring constant between 4 N/m and 14 N/m. The masses consist of hangers to which slices of weights can be attached, ranging from 5 to 20 g. The environment consisted of two hooks, each being able to hold one spring, see Figure 1. For measuring extensions and duration, we provided a measuring tape and a stopwatch.

Simulation Learning Environment. The computer simulation we used was created by PhET [22], see Figure 1. It consists of three springs, two of which have a fixed and equal spring constant. The spring constant of the third spring can be changed continuously by means of a slider. It further entails seven weights, four of which are 50g, 100g, 100g and 250g respectively. The other three have no indication of their actual weight. The weights can be attached to and removed from the springs by simple drag-and-drop. The simulation comes with a displaceable measuring tape as well as a stopwatch.

Differences in Learning Environment. Instead of designing the learning environments ourselves, we selected the ones that we considered as state of the art of their respective domains. This prevented us from setting up the necessary control of the differences in affordances of the environments for making causal claims about the relation of learning environment and experimentation strategies. However, we can reason about the potentially relevant differences based on the specific user interfaces and interaction designs. The main differences are the following ones: 1. The VME allows participants to use up to three springs, compared to two in the PME; 2. In the PME, participants could change the spring constant of both springs if needed, while the VME allowed to change the spring constant of only the third spring; 3. In the VME, manipulating the spring constant is easier as it requires only changing the value of a continuous valued slider. Participants could change its value on the fly, without interrupting an ongoing experiment. In order to change the spring constants in the PME, participants had to stop an experiment, and physically replace a spring with another one.

[1] PASCO scientific, 10101 Foothills Boulevard, P O Box 619011, Roseville, Ca 95678-9011, USA.Web: http://www.pasco.com. E-mail: sales@pasco.com. National representatives of PASCO can be reached through the USA office.

3.3.2. Subject Knowledge Assessment Questionnaire

The pre-test and the post-test consisted of four qualitative questions, each with two sub-questions. The first two questions addressed the impact of changing either the spring constant or the mass on the amplitude and frequency of oscillation. The third question targeted the understanding of force and speed in an oscillating spring-mass system. The fourth question was a near-transfer question inspired by the generalization questions of Renken & Nunez [12].

3.3.3. Procedure.

Students participated individually in the study. They were assigned randomly to either the PHY or the SIM condition. Prior to taking the pre-test, each participant was introduced to the nature and goal of the activity, and to definitions of relevant variables. Possible experiments were restricted only by the given set of weights and springs. The definition sheet contained basic definitions, both verbal and visual, of relevant concepts of harmonic oscillation of mass-spring systems. After the pre-test, the experimenter explained how to manipulate the variables and how to perform measurements, depending on condition using either the physical toolkit or the computer simulation. Participants were instructed to adjust only the settings related to the two variables of interest. They were further asked to think-aloud during the activity. The maximal duration of the inquiry task was 10 minutes. Participants then completed the post-test. Both pre-test and post-test took 5 minutes each.

3.4. Coding

3.4.1. Conceptual Tests

Pre-test and post-test items received a score of 1 if they were correctly answered, and 0 otherwise. Questions that required participants to explain their reasoning were given 0.5 for partially correct answers. The maximum score was 8. Besides the overall aggregate score, we calculated also sub-scores for the two conceptual categories, spring constant (two items) and mass dependence (two items).

3.4.2. Inquiry Behaviors

In a first pass, we extracted every experiment a participant ran from the corresponding video records of the experiment. This was done manually. Once the database was established, we could code every experiment computationally based on customized rules for

extracting relevant variables such as number of manipulated objects, etc. Even if the initial step was done by hand, the extraction procedure was operationalized such that we can automatize this process for future iterations: An experiment was characterized by the state of each relevant variable. A new experiment started when either one or more variables of the system were manipulated, or when a current experimental setup was re-initiated, either by touching a mass-spring system with the hand or with the mouse. The type of performed manipulation was then extracted from the contrast between two experiments. All variables representing inquiry behaviors are coded proportionally, relative to the total number of experiments run per activity.

An experiment consisted of the number of springs used, their spring constants, and the weights attached to the springs. The possible manipulations were (1) change of the spring constant, (2) change of the weight, (3) change both, (4) repeat an experiment, and (5) start a new experiment by changing the number of springs used. Changing either the mass only or the spring only corresponded to a *control of variables manipulation* (CVM), while a *confounded manipulation* referred to changing both variables at the same time. In cases participants used only one mass-spring configuration, we defined an experimental comparison through the contrast set up by the configurations in two consecutive runs. When two configurations were used simultaneously, the experimental comparison was defined by the contrast of those two sets of masses and springs. When participants in the SIM condition used all three springs, we defined the experimental comparison by the *most optimal* contrast out of the three possible pairwise combinations (optimal being the mass-spring configurations that differ only in one independent variable).

Table 1. Regression Models of Post-Test Scores

Variables / Models	***1***	***2***	***3***	***4***	***5***
(Intercept)	3.79***	3.12**	1.09	2.07*	2.01*
	(0.68)	(0.24)	(1.38)	(0.16)	(0.96)
Pre-test Scores	0.32†	0.32†	0.29†	0.32†	0.34*
	(0.17)	(0.17)	(0.16)	(0.16)	(0.16)
Condition	0.33	0.49	-0.05	0.38	0.36
	(0.33)	(0.44)	(0.35)	(0.37)	(0.37)
% Control of Variable		0.89			
		(1.60)			
% Confounded		1.28			
		(2.17)			
% Delib. Manip.			3.33*		
			(1.50)		
% Delib. CV				3.17*	
				(1.44)	
% Delib. Confounded				3.21	3.29
				(2.09)	(2.06)
% Delib. Spring-Only					3.95**
					(1.52)
% Delib. Mass-Only					1.46
					(1.86)
R^2	0.113	0.127	0.238	0.254	0.304
adj. R^2	0.056	0.007	0.162	0.151	0.179
N	34	34	34	34	34

Note: Standard error are in parentheses; † ($p \leq 0.1$), * ($p \leq 0.05$), ** ($p \leq 0.01$)), *** ($p \leq 0.001$); each model regresses post-test scores on the given independent variables.

As explained before, just looking at whether an experiment was unconfounded or not misses out on other relevant aspects. In particular, such a perspective does not provide any insights into how deliberately or considered participants executed and reflected on an experiment. Therefore, we additionally captured the duration of each experiment as the *dwell time* between two succeeding experimental manipulations. Based on the dwell time, we developed a *measure of deliberateness*; any manipulation that had a dwell time bigger than first quartile of all dwell times of all participants was coded as a *deliberate manipulation*.

3.5. Data Analysis.

3.5.1. Analysis of Learning Outcomes

In order to analyze the relation between inquiry behaviors and learning outcomes, we ran multiple linear regressions on post-test scores, with condition as independent factor, pre-test scores as covariate, and the corresponding measures of inquiry behavior as independent variables. For pairwise comparisons between variables within the same category that violated the normality assumptions, we report results from the nonparametric Mann-Whitney-Wilcoxon test.

3.5.2. Analysis of Inquiry Behaviors

We applied a cluster method on all experimental manipulation variables to group participants by their inquiry behaviors. We used portioning around medoids (PAM) as the clustering algorithm, which is a more robust version of the standard k-means clustering algorithm, as it minimizes a sum of dissimilarities instead of a sum of squared Euclidian distances [23]. The quality of the clustering result was evaluated based on the silhouette score [24], a measure of similarity between points and the clusters they are assigned to. The larger the silhouette value, the better the clustering. However, instead of selecting the clusters that maximize the silhouette score, we have to make a trade-off between silhouette score and number of clusters in order to have theoretically relevant results. Ideally, we could set the number of clusters to 2, as we were interested in analysis of behaviors with respect to condition.

4. Results

4.1. Baseline Knowledge

Participants in the two conditions did not differ significantly in pre-test scores, $t(32) = 1.49$, $p = 0.15$ (PHY: M = 3.53, SD = 1.59; SIM: M = 4.23, SD = 1.15). However, the high overall pre-test score average of about 52.5% of the maximal possible score indicates that participants had relevant prior knowledge with regards to the subject. We excluded two participants who scored perfectly on the pre-test. In terms of prior knowledge related to impact of the spring constant versus the mass on harmonic oscillations, there were no significant differences in pre-test scores on the corresponding subcategories (Spring constant: M = 41.2%, SD = 31.3%; Mass: M = 52.9%, SD = 30.0%), paired $t(33) = -1.54$, $d = 0.38$, $p = 0.13$. However, as the trend in data nevertheless points in the expected direction, we classify experiments that involve spring manipulations as less familiar than those involving mass manipulations.

4.2. Effect of Condition on Learning Gain

The two conditions were not significantly different in terms of average learning outcomes as condition was not a significant

factor for post-test scores, controlling for pre-test scores, $\beta = 0.33$, $t(32) = 1.01$, $p = 0.32$, $\eta_p^2 = 0.03$ (see Figure 2.B.).

4.3. Learning Outcome by Inquiry Behaviors

We examined how various measures of inquiry behaviors related to learning outcomes by multiple linear regression analysis. The baseline variables of each regression model were condition as independent factor, and pre-test score as covariate. All the corresponding regression models are shown in Table 1.

4.3.1. Time on Task and Number of Experiments

While time on task was the same across conditions, $t(32) = 0.28$, $p > 0.5$, the total number of experiments per participant was higher for the SIM condition (M = 18.7, SD = 8.3) than for the PHY condition (M = 13.7, SD = 7.3), $d = 0.64$, $t(32) = 1.87$, $p = 0.07$. Additionally, pre-test scores were not correlated with number of experiments, $r(32) = -0.05$, $p > 0.5$. An ANCOVA suggests that the number of experiments was not a significant factor for post-test scores, controlling for pre-test scores, $F(1, 30) = 0.02$, $p > 0.5$, $\eta_P^2 < 0.01$. Overall, participants performed 533 different experiments, based on which we built the database.

4.3.2. Control of Variables Manipulations

We did not find a significant effect for overall CVM on post-test scores, $\beta = 0.89$, $t(31) = 0.33$, $p > 0.5$ (see model 2 in Table 1). Even when looking at mass-only or spring-only manipulations, the respective regression coefficients are not significantly different from zero. These results indicate that performing control of variable manipulations of either the spring or the mass does not necessarily lead to better learning outcomes per se, which is in contrast to the prior literature [8]. We find that control of variable manipulations alone cannot explain the variability in learning outcomes both within and across conditions.

4.3.3. Deliberate Manipulations

We coded the deliberateness of an experimental manipulation by means of the time spend on an experiment. We extracted the duration between manipulations across all participants, and defined the cut-off value between a *rapid* and a *deliberate* manipulation as the 25th percentile of the duration histogram (Mdn = 20 seconds). This was at 11 seconds.

Overall deliberate manipulations was a relevant positive predictor of post-test scores, $\beta = 3.33$, $t(31) = 2.21$, $p = 0.03$, $\eta_P^2 = 0.14$ (model 3 in Table 1). While CVM was not relevant for learning outcomes, deliberate control of variable manipulations (DCVM) was a significant factor in the regression model 4 in Table 1, $\beta = 3.17$, $t(31) = 2.19$, $p = 0.04$, $\eta_P^2 = 0.15$. This effect was mainly driven by deliberate spring-only manipulations (see model 5 in Table 1). On the other hand, deliberate confounded manipulations had a comparably high coefficient value, even if it was not significant. With an adjusted $R^2 = 0.18$, $F(5,28) = 2.44$, $p = 0.06$, model 5 did not explain a higher proportion of variance than model 3, $F(1,2) = 1.32$, $p = 0.28$.

None of the manipulation types correlated with pre-test scores (all correlation coefficients were lower than 0.1 in absolute value). The lack of correlation supports the claim that the manipulations were context-dependent variables of inquiry behavior.

4.4. Inquiry Behavior by Condition

4.4.1. Control of Variables Manipulations and Deliberate Manipulations

The physical and the simulation condition did not differ in terms of control of variables manipulations, d = 0.14, $t(32) = -0.08$, $p = 0.94$ (SIM: M = 0.51, SD = 0.13; PHY: M = 0.53, SD = 0.16). In contrast to that, the two conditions differed significantly in the amount of deliberate control variable manipulations (DCV), $d = 0.77$, $t(32) = 2.23$, $p = 0.033$ (SIM: M = 0.35, SD = 0.15; PHY: M = 0.47, SD = 0.18). There is a significant drop in CV when considering the deliberate manipulations for the SIM condition only. In line with the hypothesis that the simulation environment was easier to manipulate, there were significantly more rapid manipula-tions in the SIM condition (Mdn = 17.6%, $CI_{.95} = \pm 24.5\%$) than in the PHY condition (Mdn = 0% , $CI_{.95} = \pm 12.9\%$), $U = 219.5$, $r = 0.46$, $p = 0.007$.

4.4.2. Cluster Analysis of Inquiry Behaviors

Overall, DCV manipulations were a significant predictor for learning outcomes, in particular the deliberate spring-only manipulations. However, even if there was a significant difference in the amount of these manipulations between the PHY and SIM conditions, learning outcomes did not differ significantly by

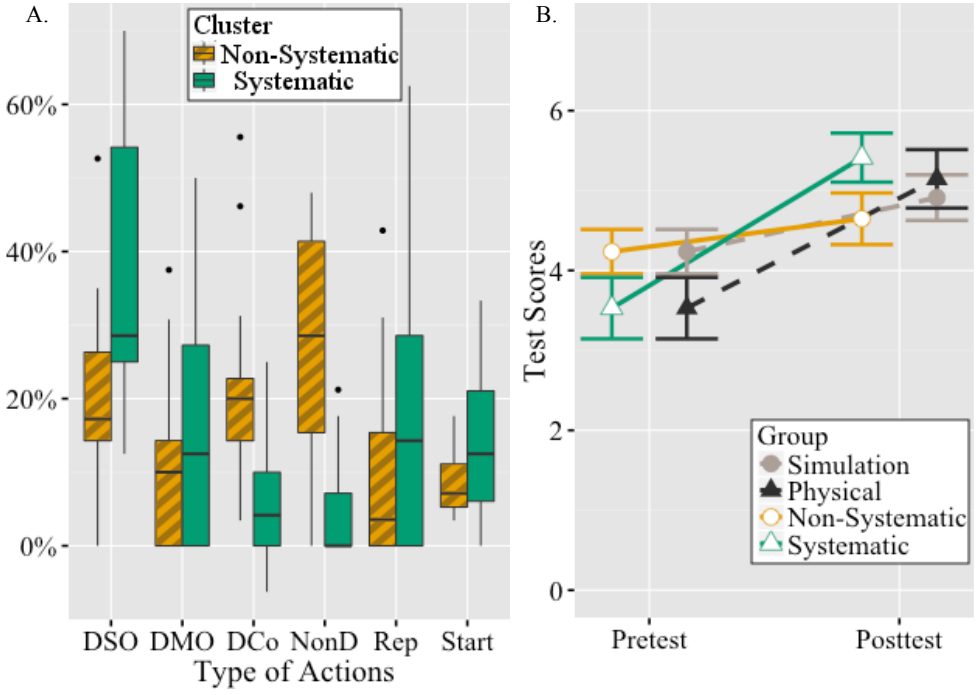


Figure 2. A. Boxplot of proportions of deliberate spring-only (DSO), deliberate mass-only (DMO), deliberate confounded (DCo), and non-deliberate (NonD) manipulations, repetitions (REP) and start of new experiments (Start). B. Comparison of pre-test and post-test scores by cluster as well as condition. Bars indicate standard errors.

condition. It appears that individual differences in inquiry strategies of participants within each condition washed out the actual impact of the learning environment on average post-test scores. There might be people in the physical and the simulation condition that deviated from the average inquiry behaviors for the condition towards the other condition's characteristics. We address this question by grouping all participants by considering all inquiry variables simultaneously instead of grouping them by condition, and then see how the groups distribute across the conditions. This can be done by means of cluster analysis.

Clustering was performed on the 6 possible manipulation types (see Figure 2.A) of the entire sample, which resulted in 2 clusters with 17 participants in each cluster. The average silhouette score was 0.30. While this score is not high enough to exclude the possibility of artificial data structures, an examination of the clusters in terms of variables confirms the clusters reasonably distinguish people by the level of systematicity of their inquiry behaviors: Generally, the participants of *Cluster 1 ("non-systematic")* were less strategic and less deliberate in their manipulations than *Cluster 2 ("systematic")* (see Figure 2.A). Cluster 2 had a higher proportion of deliberate spring-only manipulations than Cluster 1, $U = 58$, $r = 0.51$, $p = 0.002$, a lower proportion of non-deliberate manipulations than Cluster 1, $U = 262.5$, $r = 0.72$, $p < 0.001$, and a lower proportion of confounded manipulations, $U = 240.0$, $r = 0.57$, $p < 0.001$. There was no significant difference in the other variables. Additionally, even if the clustering was not performed on overall DCV, there is a large difference between the clusters; participants in the systematic cluster proportionally performed significantly more DCV (Mdn = 49.8%, $CI_{.95} = \pm 16.3\%$) than in the non-systematic cluster (Mdn = 30.7%, $CI_{.95} = \pm 12.1\%$), $d = 1.33$, $t(32) = 3.89$, $p < 0.001$.

The two clusters meaningfully differ in learning outcomes, as indicated by a regression of post-test scores on the cluster variable, with pre-test scores as covariates, which revealed a significant main effect of cluster, $\beta = 1.03$, $t(31) = 2.39$, $p = 0.015$, $\eta_P^2 = 0.16$. As expected, participants in the systematic scored higher than those in the non-systematic cluster (see Figure 2.B). The regression model explained a significant proportion of variance, adjusted $R^2 = 0.18$, $F(2,31) = 4.55$, $p = 0.02$.

Table 2. Conditions distributed across clusters

Condition	Non-Systematic (n = 17)	Systematic (n = 17)
Physical (n = 17)	*3 (17.6%)*	*14 (82.4%)*
Simulation (n = 17)	*14 (82.4%)*	*3 (17.6%)*

Finally, Table 2 shows that the majority of participants in the systematic cluster used the physical toolkit, while the majority of participants that belonged to the non-systematic cluster were in the simulation condition, as confirmed by a Fisher's exact test, $p < 0.0001$.

5. DISCUSSION

Considerable attention has been given separately to research on the impact of virtual and physical learning environment [4] and of inquiry behaviors on the learning outcomes in science discovery activities [8,9]. The aim of the present study was to link these two realms by (1) studying the relation of strategy use and learning outcomes, and (2) comparing strategy use between learning environments in order to shed light on how different affordances of the learning environments might influence strategy use.

5.1. Nuanced View of Experimentation Strategies in Open-Ended Inquiry Tasks

One main finding from this study was that one of the strongest predictors for learning outcomes when controlling for prior knowledge was the manipulation type that (a) created a single contrast in experiment conditions, (b) targeted the problem type that participants generally were less familiar with, and (c) was deliberate. In the context of the mass and spring activity, these were deliberate manipulations that changed only the spring constant from one mass-spring system to the other.

Importantly, this further implies that the control of variables (CV) in experiment design was a necessary but not sufficient condition for developing conceptual understanding through experimentation. This is in contrast to prior research that has predominantly focused on the ability to design unconfounded experiments as the main factor of knowledge acquisition in inquiry learning [2,10,12]. Using control of variable strategy as an important factor for characterizing experimentation strategies works when the student has to make a conscious decision to actually apply this strategy. It fails if the affordances of the user interface do not require that. In the computer simulation, one could change the spring constant continuously using a slider, even during an ongoing experiment. In the physical condition however, an experiment had to be interrupted in order to change either the mass or the spring, which required the participant to deliberately decide what to manipulate, but both changes are coded as CV manipulations. As a consequence, we not only found that there was no difference in CV manipulations between conditions, but also that these manipulations did not have predictive value for learning outcomes.

This picture changed when accounting for the *deliberateness* of experimental manipulations. It turned out to that in contrast to CV manipulations, the percentage of deliberate CV manipulations significantly predicted learning outcomes, as well as differed between conditions. The drop from CV to deliberate CV manipulations was significant only for the SIM condition. This is in line with our reasoning that the user interface for the computer simulation did not make the control of variables a deliberate choice. Even by itself, deliberate manipulations were among the strongest predictor for post-test scores. We suggest that time between manipulations as a measure of deliberateness is not just reflective of the ease of manipulation in a learning environment, but also of the level of cognitive engagement of a participant with an experiment.

Finally, only manipulations targeting the less familiar concept (spring) contributed to conceptual learning, while those targeting the more familiar one (mass) did not seem to impact the learning outcomes, which seems reasonable given that the participants tended to know less about the springs' role in the harmonic oscillation. However, contrary to previous studies [12] that consider confounded manipulations as detrimental to developing conceptual understanding, we found a relatively large though insignificant positive regression coefficient for confounded manipulations on post-test scores. At this point, we can only speculate as to why this is the case; for example, it could be that people with low prior knowledge ran preliminary experiments to get a sense of the physical phenomenon. Further investigation is needed to understand this process.

5.2. Differences in Inquiry Behaviors by Learning Environment

We found that conditions did not differ in terms of learning outcomes. In line with previous research that showed equal knowledge gains for virtual and physical manipulative environments [2, 3, 5, 7], we could have argued that there is no difference in benefits of learning environments for developing conceptual understanding in inquiry tasks on mass-spring systems. However, as indicated by the results of the cluster analysis of inquiry behaviors, this would have been the wrong conclusion. The cluster analysis revealed that participants across both conditions could be grouped into two clusters according to how systematic their inquiry behavior was, and that the more systematic cluster had significantly higher learning outcomes than the less systematic cluster. Importantly, almost all of the participants in the physical condition belonged to the more systematic cluster, while most of the participants in the simulation condition fell into the less systematic cluster. This suggests that the learning environments did differ in terms of benefits for developing conceptual understanding. It is important to note that this is not in contradiction to the multiple regression models that show no significant effect for condition. Both analyses show that inquiry strategies had a strong influence on learning outcomes. However, enough participants deviated from their peers in the same condition in terms of inquiry behaviors such that the overall differences in learning outcomes between conditions were canceled. By using more than one variable of inquiry behavior for grouping participants, cluster analysis better accounts for between subject differences in overall inquiry behaviour in each condition. Thus, at least for activities that span a short period of time, we think that measures of experimentation strategies have to be incorporated in studies of the impact of learning environments on learning outcomes in open-ended science inquiry learning.

A possible explanation for these differences in experimental manipulations between conditions is that the ability to employ systematic experimentation strategies is not necessarily a stable domain-general skill but a context-dependent behavior. It is likely that specific affordances of the two learning environments are related to these differences in experimentation strategies, such as the need to pause the experiment to change the spring constant in the real but not virtual environment. While there is consensus on the impact of different affordances of virtual and physical environments on learning outcomes [4], we argue in light of these results that we also need to study the impact of these affordances on the experimentation processes during science inquiry activities. However, as we did not manipulate the specific affordances in the learning environments, we can currently only make educated guesses.

For example, the fact that participants in the SIM condition ran more experiments than in PHY, while spending the same amount of time at the task, supports the claim that it was easier to manipulate variables in the computer simulation than in the physical setup. As argued by Renken and Nunez [12], it might be that systems that enable quick changes with various options prompt participants to get into "play" mode, in which they revert to simple heuristic methods such as trial-and-error and spend less effort on setting up valid experiments. This could explain why proportion of deliberate manipulations was higher for participants using the physical systems.

Another difference in affordances is that in the computer simulation, participants could change the spring constant even as experiments were running, which led to short perturbations in the oscillations that were due to the change, and not necessarily due to the actual spring-mass configurations. Especially in cases "non-deliberate" manipulations that were too short for the perturbations to vanish, participants might have wrongly interpreted these fluctuations.

5.3. Limitations and Future Directions

While the study provided evidence that an investigation of inquiry strategies is more informative than merely looking at outcomes, it only offered hints as to what determines the use of those strategies. These appear to be influenced by the different affordances of a learning environment, but studies with longer interaction times, and a greater range and control of environments is needed to understand the characteristics of these relationships in more detail. Future studies should better control and match the virtual and physical environments in order to focus on one or two specific affordances. Studies that manipulate design features *within* a learning environment to assess its impact on inquiry processes are also needed.

Further studies should incorporate the assessment of hypothesis generation and inference processes to examine the impact of affordances of learning environments not just on experimentation strategies, but on these other critical inquiry behaviors as well.

We found that time between manipulations was an important correlate of learning outcomes; however, wit the current study, we can make only educated guesses as to what cognitive processes longer dwell times correspond to. Dwell time could signify the time spent on comparing the current with the prior experiment configuration, on reflecting on existing confusions, on planning the next steps to be taken, or it could just represent the time it takes to perform a manipulation in the learning environment.

Additionally, the lack of difference on learning outcomes between media seems to contradict prior research on virtual versus physical learning environments in comparable inquiry tasks [12]. However, as the tendency of the data goes into the expected direction, we believe that a larger sample size would provide the required power to detect the learning outcome differences.

We currently did not employ automated tracking of participants' behaviors to extract their experiment configurations. However, novel computer vision algorithms, as well as logging systems would address this limitation. Our data organization scheme can be easily integrated with automatized tracking systems.

6. CONCLUSION

Drawing on work on scientific reasoning and inquiry, we developed a novel operationalization of systematic experimentation strategies that predict learning outcomes in open-ended inquiry-based learning activities. We further showed that strategy use is context-dependent, in that participants using the physical system went about the inquiry activity differently than participants using the computer simulation.

These findings suggest that we have to broaden the notion of what counts as "systematic experimentation" from mainly consisting of the design of unconfounded experiments and the performance of optimal heuristic search to a more comprehensive views that integrates contextual and cognitive factors (e.g. deliberateness). Data mining algorithms are particularly well suited for exploring such behaviors. However, it is crucial to develop data-mined models of inquiry strategies that are interpretable in order to advance our understanding of learning processes in more complex

inquiry activities. We suggest that any machine-learned model of inquiry behaviors should incorporate semantic representations of what participants' actually explore in inquiry activities, in order to meaningfully extend the data from interaction logs of users engaging in the learning environment.

A further implication of our results is that research on learning environments for science inquiry learning should focus on developing a broader framework that focuses on the affordances as relevant dimensions, irrespective of medium and examines how under what circumstances they benefit learning.

7. ACKNOWLEDGEMENT

We would like to thank Prof. Carl Wieman and Eric Kuo, PhD, for their guidance and strong support in this research, as well as members of the AAALab at the Stanford University for their insightful feedback.

8. REFERENCES

[1] van Joolingen, W., & Zacharia, Z. (2009). Developments in inquiry learning. In *Technology-enhanced learning.* Netherlands: Springer.

[2] Triona, L., & Klahr, D. (2003). Point and click or grab and heft: Comparing the influence of physical and virtual instructional materials on elementary school students' ability to design experiments. *Cognition and Instruction, 21*, 149-173.

[3] Zacharia, Z., & Olympiou, G. (2011). Physical versus virtual manipulative experimentation in physics learning. *Learning and Instruction , 21*, 317-331.

[4] de Jong, T., Linn, M., & Zacharia, Z. (2013). Physical and virtual laboratories in science and engineering education. *Science, 340*, 305-308.

[5] Klahr, D., Triona, L., & Williams, C. (2007). Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching, 44*, 183-203.

[6] Zacharia, Z., & Constantinou, C. (2008). Comparing the influence of physical and virtual manipulatives in the context of the physics by inquiry curriculum: The case of undergraduate students' conceptual understanding of heat and temperature. *American Journal of Physics, 76*, 425-430.

[7] Pyatt, K., & Sims, R. (2012). Virtual and physical experimentation in inquiry-based science labs: Attitudes, performance and access. *Journal of Science Education and Technology, 21* (1), 133-147.

[8] Zimmerman, C. (2000). The development of scientific reasoning skills. *Developmental Review, 20*, 99-149.

[9] Zimmerman, C. (2007). The development of scientific thinking skills in elementary and middle school. *Developmental Review, 27*, 172-223.

[10] Chen, Z., & Klahr, D. (1999). All other things being equal: Acquisition and transfer of the control of variables strategy. *Child Development, 70*, 1098-1120.

[11] Klahr, D. (2000). *Exploring science: The cognition and development of discovery processes.* Cambridge: MIT Press.

[12] Renken, M., & Nunez, N. (2013). Computer simulations and clear observations do not guarantee conceptual understanding. *Learning and Instruction, 23*, 10-23.

[13] Shih, B., Koedinger, K., & Scheines, R. (2010). Unsupervised Discovery of Student Strategies. *Proceedings of the 3rd Intl. Conf. on Educational Data Mining*, (pp. 201-210).

[14] Kardan, S., & Conati, C. (2011). A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. *Proceedings of the 4th Intl. Conf. on Educational Data Mining*, (pp. 159-168). Eindhoven, the Netherlands.

[15] Kardan, S., Roll, I., & Conati, C. (2014). The usefulness of log based clustering in a complex simulation environment. *Intelligent Tutoring Systems* (pp. 168-177). Springer International Publishing.

[16] Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., & Nakama, A. (2013). Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction, 23* (1), 1-39.

[17] Sao Pedro, M., Baker, R., & Gobert, J. (2012). Improving construct validity yields better models of systematic inquiry, even with less information. *User Modeling, Adaptation, and Personalization* (pp. 249-260). Berlin Heidelberg: Springer.

[18] Palmer, D. (1995). *The POE in the primary school: An evaluation. Research in Science Education*, 25 (3), 323-332.

[19] Penner, D., & Klahr, D. (1996). The interaction of domain-specific knowledge and domain-general discovery strategies: A study with sinking objects. *Child Development, 67*, 2709-2727.

[20] Schauble, L. (1996). The development of scientific reasoning in knowledge-rich contexts. *Developmental Psychology , 32*, 102-119.

[21] Garcia-Mila, M., & Andersen, C. (2007). Developmental change in notetaking during scientific inquiry. *International Journal of Science Education, 29* (8), 1035-1058.

[22] Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., & LeMaster, R. (2006). PhET: Interactive simulations for teaching and learning physics. The Physics Teacher, *44*(1), 18-23.

[23] Reynolds, A., Richards, G., de la Iglesia, B., & Rayward-Smith, V. (1992, 5). Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 475-504.

[24] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, 53-65.

Toward a Real-time (Day) Dreamcatcher: Sensor-Free Detection of Mind Wandering During Online Reading

Caitlin Mills
University of Notre Dame
Department of Psychology
Notre Dame, IN 46556
cmills4@nd.edu

Sidney D'Mello
University of Notre Dame
Department of Psychology
Department of Computer Science
Notre Dame, IN 46556
sdmello@nd.edu

ABSTRACT
This paper reports the results from a sensor-free detector of mind wandering during an online reading task. Features consisted of reading behaviors (e.g., reading time) and textual features (e.g., level of difficulty) extracted from self-paced reading log files. Supervised machine learning was applied to two datasets in order to predict if participants were mind wandering as they navigated from one screen of text to the next. Mind wandering was detected with an accuracy of 20% above chance (Cohen's kappa = .207; AUC = .609), which was obtained via leave-one-participant-out cross-validation. Similar to actual rates of mind wandering, predicted rates of mind wandering were negatively related to posttest performance, thus providing some evidence for the predictive validity of the detector. Applications of the detector to attention-aware educational interfaces are discussed.

Keywords
Mind wandering, attention, machine learning, reading

1. INTRODUCTION
It is not uncommon to experience looking up from a book only to realize you have no idea what you just read. In fact, it has been documented that people can read up to 17 words of gibberish before even realizing that they have zoned out [32]. Since students often have trouble realizing when they have zoned out themselves, it can be especially difficult to determine when someone is not paying attention through observation. For example, a student who is deeply engaged in learning can often look quite similar to another student who is thinking about something else completely.

This phenomenon, known as *mind wandering*, is an involuntary shift in attention away from the external task towards task-unrelated thoughts [36]. Mind wandering is detrimental during learning, as learning requires consolidating external information into mental structures. During episodes of mind wandering, however, students are unable to integrate external information with their existing internal representations. Thus, missed information is not processed and mental models are not updated, limiting overall understanding. Given the negative impact of mind wandering on learning [14, 30, 32, 33], it is important to develop systems that can reorient attention when students mind wander in order to facilitate engagement and learning. Building detectors of mind wandering is an essential first step towards this goal and is the focus of the present paper.

1.1 Related Work
One of the first known studies related to mind wandering detection was conducted by Drummond and Litman [13]. In their study, students read a paragraph about biology aloud then performed a learning task (i.e., paraphrase or self-explanation). Students periodically self-reported how frequently they were thinking about off-task thoughts on a scale from 1 (all the time) to 7 (not at all). Supervised machine learning trained on acoustic-prosodic features was used to classify whether students were "high" in zoning out (1-3 on the scale) versus "low" in zoning out (5-7 on the scale). Results indicated an accuracy of 64% in discriminating "low" versus "high" zone outs. This pivotal study on mind wandering was innovative with respect to automatically detecting zone outs during a learning task. However, they used a leave-one-instance-out cross-validation method (rather than a leave-one-participant-out cross-validation method), so generalizability of the detector to new students is unclear.

Recent research has also attempted to detect mind wandering during online reading using both gaze [5] and peripheral physiology [6]. In both of these studies, mind wandering was collected via thought probes that occurred on pseudo-random pages (i.e., computer screens) during reading. Students responded either "yes" or "no" about whether they were mind wandering at the time of the probe. In the first study, a detector of mind wandering achieved an accuracy of 72% (Cohen's kappa = .28) using features extracted from gaze data collected with a Tobii eye tracker [5]. In the second study, a detector of mind wandering built using physiological features (i.e., skin conductance and temperature) achieved an accuracy of 74% (Cohen's kappa = .22). Both of these detectors used a leave-several-subjects-out validation method to ensure generalizability to new students.

These detectors display impressive results given the elusive nature of mind wandering. However, the equipment and sensors required for eye-gaze and physiology tracking might impair scalability. In particular, one issue faced by online learning environments is that sensors are not readily available. For example, students using an ITS deployed online from their home computer would not have access to an eye tracker or a way to measure skin conductance at their convenience. A key question then is how to detect mind wandering based on information that is readily available, for example, in interaction log files. Along these lines, the aim of the current study is to identify a set of features that 1) are theoretically

related to mind wandering, and 2) can be extracted from log files during online learning.

Interaction-based detectors trained from interaction log files have been used to successfully build detectors of other "off-task" states, such as gaming the system and off-task conversation [4, 7–9]. While mind wandering is related to other forms of "off-task" states, such as boredom, behavioral disengagement, and distractions [1, 3, 4, 8, 9, 26, 42], it is inherently distinct because it is involuntary and involves internal thoughts rather than overt expressive behaviors. The involuntary, unconscious nature of mind wandering makes detection particularly difficult. First, whereas other off-task states often involve some behavioral markers to denote disengagement, mind wandering is a completely internal state that can look similar to on-task states. Second, the onset and duration of mind wandering episodes cannot be precisely measured because people are often unaware their attention has been directed away from the external task. Thus, finding features that will pick up on subtle differences in attention is extremely difficult.

To date, one study has attempted sensor-free mind wandering detection (see Table 1 for a summary of mind wandering detectors). Franklin et al. [15] attempted to classify if readers were "mindlessly reading" using two criterion: (1) difficulty and (2) reading time. For the first criterion, readers could only be classified as "mind wandering" while reading difficult text. To establish the level of difficulty, each word was assigned a difficulty score based on the average of three binary ratings: (1) length (at least four letters = 1, less than four letters = 0), (2) syllables (at least two syllables = 1, under two syllables = 0), and (3) familiarity (based on a psycholinguistic database where above average = 1, below average = 0). Then, the average difficulty across a running window of 10 words had to be above a threshold set at .45 for a reader to be classified as "mindless reading." The second criterion was based on reading time. Participants read one word on a screen at a time. Using a running window of 10 words, a specific threshold (based on pilot data) was applied to determine when readers were reading either too fast or too slow.

Table 1. Overview of Previous Mind Wandering Detectors

	Key Features	Classification Accuracy	Validation Method
Bixler et al. (2014)	Eye Gaze	72% correct	leave-several-subjects-out
Blanchard et al. (2014)	Physiology	74% correct	leave-several-subjects-out
Drummond et al. (2010)	Prosodic/ Lexical	64% correct	leave-one-instance-out
Franklin et al. (2011)	Difficulty/ Reading Time	72% correct	thresholds derived from pilot data

This study provided some evidence that reading time, combined with textual features such as difficulty, might be indicative of mind wandering (accuracy = 72%). However, since reading times were collected by presenting one word on the screen at a time, their methods and predetermined thresholds for fast and slow reading may not be generalizable to other, more natural, reading contexts. Additionally, mind wandering was never predicted to occur during "easy" portions of the text, which may not accurately reflect the real-life occurrence of this phenomenon. For example, mind wandering still occurs around 20% during easy texts [27], even though it is more frequent during difficult texts. Furthermore, their method relied on a number of pre-set thresholds with little information on how these thresholds were established, thereby complicating attempts to replicate their results.

1.2 Current Study

This paper reports a person-independent detector of mind wandering during a more natural, computerized self-paced reading task using basic information that can be extracted from reading logs. In an attempt to provide a foundation for an easily-scalable way to capture when mind wandering occurs, the detector is completely sensor-free.

The mind wandering detector was trained on two unpublished datasets in which participants attempted to learn about scientific research methods by reading texts presented online. Participants completed a posttest after reading in order to assess learning. Importantly, these datasets include diversity with respect to population, methods, and level of text difficulty. For example, dataset 1 was collected via Mechanical Turk, a validated online data collection platform [23], and had an average age of 35 years. Dataset 2 was collected from a Midwestern university subject pool and had an average age of 19 years. Therefore, building a detector of mind wandering using more than one dataset with varying conditions will increase our confidence in its relative generalizability.

2. DATASETS

The datasets were originally collected to investigate mind wandering under various conditions, such as varying levels of difficulty and text presentations. In addition, a posttest was completed after reading in order to assess how mind wandering relates to learning. In both datasets, participants were instructed to read the text carefully and notified that they would be asked to answer questions about content from the text after reading. Dataset 1 (N = 177) was collected on Amazon's Mechanical Turk, an online data collection platform that has been validated for high quality data [23, 28]. Participants were compensated $2.50 after completing the experiment. Dataset 2 (N = 141) was collected via an online subject pool at a Midwestern university in the United States. Participants received class credit after completing the study.

Table 2 provides an overview of the experimental designs and manipulations used in each dataset. The Text Difficulty manipulation involved participants reading texts that were experimentally manipulated to be either "easy" or "difficult" (see section 2.1 for manipulation details). The Text Presentation manipulation involved participants reading either one sentence or one paragraph at a time on the screen.

2.1 Reading Materials

The two texts used in the existing datasets were adapted from texts used in the serious game, *Operation ARA!* [25]. Each text focused on a concept related to research methods: (1) the dependent variable and (2) making causal claims, both of which are key concepts relevant to understanding the scientific method. In the existing datasets, easy and difficult versions of each text

were used in order to investigate the effect of text difficulty on mind wandering.

Easy versions of the text were more narrative in nature, and consisted of shorter sentences and fewer low frequency words (average Flesh-Kincaid Grade Level = 9). Difficult versions of the text consisted of longer, more complex sentences with more low frequency words (average Flesh-Kincaid Grade Level = 13). Both versions had the same conceptual content and were approximately 1500 words in length. An example of an easy sentence is, "People who know about the scientific method do not fall for unsupported claims like this one." The difficult version of the same sentence was, "So many citizens fall for these dubious claims, but people who comprehend the scientific method are not victimized by these unsupported claims."

2.2 Procedure

Participants first completed an electronic consent form. They were then given instructions for the self-paced learning task. Participants pressed the space bar to move through each screen of the text. Texts were presented on screen either one sentence at a time or one paragraph at a time based on experimental manipulation (see Table 2).

Mind wandering was tracked via auditory thought probes in both datasets. A standard description of mind wandering [36] was employed: "At some point during reading the texts, you may realize that you have no idea what you just read. Not only were you not thinking about the text, you were thinking about something else altogether." The probe consisted of an auditory beep that occurred on pseudo-random screens throughout each text. Probes were triggered when participants pressed the space bar to advance to the next portion of the text. Participants were instructed to press the "Y" key if they were mind wandering or the "N" key if they were not. Participants were not able to advance to the next screen until they had responded to the mind wandering probe. A total of six auditory mind wandering probes were inserted in each text. Probes were placed in an identical location with respect to content within each text. That is, regardless of whether the text was presented one sentence or paragraph at a time, the probe would occur after reading identical content.

Table 2. Overview of Two Datasets

	Dataset 1	**Dataset 2**
Sample	Mechanical Turk	University subject pool
# Texts	1	2
# Participants	177	141
Manipulations:		
Text Difficulty	Easy/Difficult	Difficult only
Text Presentation	Par/Sen	Par/Sen

Notes. Par = Paragraph-by-paragraph; Sen = sentence-by-sentence

Participants completed a posttest after reading each topic. Posttests consisted of four-alternative multiple-choice questions that tapped two levels of comprehension: (1) surface level, and (2) inference level. Surface level questions were based on factual or text level characteristics of the text. Inference questions were designed to elicit patterns of reasoning and require participants to use inference or apply a learned concept to a novel example in order to answer the question correctly [19]. For dataset 2, participants answered an 18-item posttest that covered both topics, which included six inference and 12 surface level multiple-choice questions. Since only one text was read during dataset 1, the posttest was limited to the 9 corresponding questions (3 inference and 6 surface level questions).

2.3 Mind Wandering Reports

Every screen of text where a probe was triggered was classified as either "Mind Wandering" or "Not Mind Wandering" based on participants' response to the probe. The two datasets were pooled in order to maximize training and validation data. In total, there were 2754 probe screens that were used to build the models. Participants indicated they were mind wandering in response to 31.3% of all the probes. Thus, our data set contained 861 instances of Mind Wandering and 1893 instances of Not Mind Wandering.

3. MODEL BUILDING

3.1 Feature Engineering

A considerable amount of empirical research has been dedicated to understanding mind wandering through experimental manipulations, such as comparing mind wandering across various conditions. Other studies have focused on explaining the behavioral correlates and temporal patterns of mind wandering [14, 16, 16, 27, 34, 38, 40]. The features in the current research were informed by the following discoveries about mind wandering: First, mind wandering is affected by the difficulty of a task [14, 27]. Second, mind wandering is related to response times and lexical features [15, 29]. Third, mind wandering rates vary as a function of time on task [30, 40]. In line with these findings, a total of 13 features were computed based on information that can found in log files. The 13 features can be subdivided into three categories: (1) Reading Behavior Features (2 features), (2) Textual Features (8 features), and (3) Context Features (3 features).

Reading Time Features. Participants' reading time (i.e. how long they spent on each screen) was collected during the reading task. Importantly, the thought-probe was triggered as participants attempted to move on to the next screen. Therefore, we can use reading behaviors from the current screen of text (screen K) to detect whether they are mind wandering or not before they moved on to the next screen (K+1).

The first reading behavior feature was *Reading Time*, which was simply the amount of time spent reading a given paragraph before pressing the space bar to advance onto the next screen. Reading Time was computed at the paragraph level in order to account for differences in reading times across the Text Presentation manipulation. When texts were presented one paragraph at a time, *Reading Time* was simply how long they spent on the screen leading up to the thought-probe. When texts were presented one sentence at a time, sentences were aligned with the content from the paragraph presentation condition. Thus, *Reading Time* was calculated as the amount of time spent reading identical content before the thought-probe regardless of presentation style.

The second reading behavior feature was called *Decoupling* [41]. *Decoupling* is a theoretically-driven metric based on the idea that reading times should increase with more complex text characteristics, such as sentence length and other discourse features [18]. If participants are not appropriately allocating resources (i.e., increasing reading times when text complexity increases) to meet the current task demands, then we might expect deviation from this linear relationship thus indicating decoupling

from the reading task. *Decoupling* was computed on the alignment (or misalignment) of reading times and text complexity. Text complexity was assessed using Flesh-Kincaid Grade Level (FKGL; [22]). The formula used to calculate decoupling was: |*z*-score standardized reading times – *z*-score standardized FKGL|. It is important to point out that decoupling was computed using the absolute value of the difference between reading time and text complexity, such that higher values would occur both when reading times were both over and under appropriated relative to text complexity. Thus, we are primarily interested in how well the overall magnitude of deviation in the relationship between reading time and text complexity can predict mind wandering.

Textual features. Eight textual features were computed in total. The first feature was simply the *Number of Characters* in the current paragraph. The second feature was the *Number of Words* in the current paragraph. Both features were used because they may differ notably between easy and difficult conditions, as easy texts were specifically manipulated to contain shorter words. Regardless of whether the screen was being presented one paragraph at a time or one sentence at a time, these features were used to represent the length of the current unit of text being processed. Longer paragraphs may require increased cognitive resources (related to mind wandering [24]) when a single idea must be kept in working memory across larger amounts of text. The third feature was *FKGL* [22], an indicator of reading level that is derived from the number of syllables and word length in a sentence. The current FKGL was also computed based on the current paragraph being read, as this metric is not reliable for extremely small portions of text, such as a single sentence.

The remaining five textual features were computed using Coh-Metrix, a program that analyzes texts across multiple levels of cognition and comprehension [17, 18]. We used five different features from Coh-Metrix: (1) Narrativity, (2) Deep Cohesion, (3) Referential Cohesion, (4) Syntactic Simplicity, and (5) Word Concreteness. *Narrativity* is computed based on how well the text aligns with the narrative genre, by conveying a story, procedure, or sequence of actions. *Deep Cohesion* is computed based on how well different ideas in the text are cohesively tied together in order to signify causality or intentionality. *Referential Cohesion* is based on how words and ideas are connected to each other across the span of the story or text. *Syntactic Simplicity* is computed based on the simplicity of the syntactic structures in the text. Lastly, *Word Concreteness* is based on the degree to which context words evoke concrete mental images, rather than abstract or conceptual representations.

Context features. Three context features were also computed based on the context of the reading task. *Current Paragraph Number* is the number of paragraphs read from the beginning of the text. *Current Difficulty* is whether the text was experimentally manipulated as easy or difficult. *Current Presentation* is whether the text was being presented one sentence at a time or one paragraph at a time.

3.2 Supervised Classification and Validation

We used supervised machine learning to build detectors of mind wandering for each screen that included a thought-probe. The goal of the paper was to create a detector that would accurately predict whether participants responded "yes" or "no" to the mind wandering probes. RapidMiner, a popular machine learning tool, was used to train binary classifiers to make this distinction. In total, four binary classifiers provided in RapidMiner were used, including Naïve Bayes, Bayes Net, RIPPER (JRip implementation), and C4.5 (J48 implementation). Down-sampling was used to create equal classes for the training data only. This was achieved by randomly selecting 45.4% of the Not Mind Wandering instances and 100% percent of the Mind Wandering instances for training. The original distributions were not changed in the testing data to preserve the validity of the results.

Manual feature selection was applied by removing one feature at a time and assessing performance on held-out testing data (see below). If model performance decreased after a feature was removed, it was preserved for the final model[1].

All models were evaluated using leave-one-participant-out cross-validation, in which *k*-1 participants are used in the training data set. The model was then tested on the participant who was not used in the training data. This process was repeated *k* times until every participant was used as the testing set once. Cross-validating at the participant level increases confidence that models will be more generalizable when applied to new participants because the testing and training sets are independent.

Classification accuracy was evaluated using two metrics: (1) Area Under the ROC Curve (AUC), and (2) Cohen's kappa. AUC is statistically similar to A′ [21] and ranges from 0 to 1, where 0.5 is chance level of accuracy and 1 is perfect accuracy. Cohen's kappa [10] indicates the degree to which the model is better than chance (kappa of 0) at correctly predicting Mind Wandering or Not Mind Wandering. A kappa of 1 indicates the detector performs perfectly. We also report percent correctly classified (accuracy), but note that this should be interpreted cautiously since class imbalance tends to inflate accuracy.

4. RESULTS

4.1 Classification Accuracy

Four classification algorithms (J48, JRIP, Naïve Bayes, and Bayes Net) were applied to the two combined datasets. The final models reported in this section were selected based on the highest AUC achieved after testing all four classification algorithms. A final combined feature model (combined model) was achieved with the J48 decision tree classifier using six features from the feature subtypes: *Reading Time*, *Decoupling*, *Number of Characters*, *Number of Words*, *FKGL*, and *Referential Cohesion.* Importantly, the combined model performed at rates above chance (AUC = .609; kappa = .207; accuracy = 63%). Despite using information solely obtained from log files and text characteristics, these accuracy rates are only slightly lower than the sensor-based detectors of mind wandering reported in Table 1.

We also examined the confusion matrix for the final combined model (see Table 3). The model had a relatively high rate of misses (.427), where actual instances of Mind Wandering were predicted as Not Mind Wandering. However, the model also displayed more correct rejections (.653), such that Not Mind Wandering instances were accurately classified as Not Mind Wandering. This was complemented by a low rate of false alarms as well (.347).

We were also interested in exploring how each of the three feature subtypes (i.e., reading behaviors, textual, and context features) were able to predict mind wandering independently. Each group of feature subtypes was therefore tested independently using the same four classification algorithms (J48, JRIP, Naïve Bayes, and Bayes Net). A summary of the classification accuracies for the

1 We also tested models using all 13 features, which exhibited lower performance (assessed via AUC) than the combined model using feature selection.

best performing models (selected based on highest AUC) can be found in Table 4.

Table 3. Confusion Matrices of Combined Model

	Pred. MW	**Pred. Not MW**	**Priors**
Actual MW	.573 (hit)	.427 (miss)	.313
Actual Not MW	.347 (false alarm)	.653 (correct rejection)	.687

Note. Pred. = Predicted; MW = Mind Wandering

All three models built from the feature subtypes performed above chance levels (AUC > .5). However, none of these models performed as well as the combined model. For example, the Textual Features Only model did not perform as well in the absence of reading time behaviors and vice versa. This suggests that using a range of feature types might help with classification accuracies rather than a subset of features.

Based on the confusion matrices, it appears that the three feature subtype models exhibited different patterns of classification (see Table 5). Although the Reading Behaviors Only model (*Reading Time* and *Decoupling*) displayed the lowest hit rates (.439), this model also had the highest rate of correct rejections. Conversely, the Textual Features Only (five Coh-Metrix dimensions, *Number of Characters*, and *Number of Words*) and the Context Features Only (*Current Presentation*, *Current Difficulty*, and *Current Paragraph Number*) models had similar higher hit rates, but fewer correct rejections compared to the Reading Behaviors Only Model.

Table 4. Performance Metrics

Features in model	**AUC**	**Kappa**	**Classifier**
Combined Model	.609	.207	J48
Reading Behaviors Only	.560	.122	J48
Textual Features Only	.591	.115	Bayes Net
Context Features Only	.542	.104	JRIP

It is important to point out that the combined model's confusion matrix also shared some similarities with the feature subtype models. The Reading Behavior Only model had the highest correct rejections (.687), which were on par with the combined model (.653). Similarly, the Textual Features Only and Context Features Only models had the best hit rates (.554 and .557), which were also on par with the hit rates in the combined model (.573). Thus, the combined model appears to strike a balance between hits and correct rejection, which is why it yields the highest AUC compared to the individual models.

4.2 Feature Analysis

Since our features were modeled after empirically-supported relationships of mind wandering (see Section 3.1), we explored how our features related to the model's predictions of mind wandering. For each participant, we computed the mean of each feature as well as the proportion of predicted mind wandering (based on the combined model's predictions). As an additional step, the averages were *z*-score standardized across the two datasets to account for the differences in methods. Predicted mind wandering was then regressed on each of the six features included in the combined model, $F(6,317) = 35.5$, $p < .001$, $R^2_{adjusted} = .395$. The regression allowed us to examine the relationship between each of the features and predicted mind wandering while controlling for the other features in the model. Table 6 presents a summary of the features used the combined model, as well as the standardized regression coefficient (β) for each feature.

Table 5. Confusion Matrices for Each Feature Set Separately

Reading Behavior	*Pred. MW*	*Pred. Not MW*
Actual MW	.439 (hit)	.561 (miss)
Actual Not MW	.313 (false alarm)	.687 (correct rejection)
Textual Features	*Pred. MW*	*Pred. Not MW*
Actual MW	.554 (hit)	.446 (miss)
Actual Not MW	.424 (false alarm)	.576 (correct rejection)
Context Features	*Pred. MW*	*Pred. Not MW*
Actual MW	.557 (hit)	.443 (miss)
Actual Not MW	.432 (false alarm)	.568 (correct rejection)

Note. Pred. = Predicted; MW = Mind Wandering

Reading Time was negatively related to predicted mind wandering, indicating that mind wandering predictions were associated with faster reading times. The second reading behavior feature, *Decoupling*, was positively related to predicted mind wandering. Mind wandering was more likely to be predicted when decoupling scores were higher, since higher decoupling scores indicate a misalignment between reading times compared to text complexity.

Number of Characters and *Number of Words* were both positively related to predicted mind wandering, suggesting that more content in general is associated with greater predictions of mind wandering. This is also related to the idea that longer paragraphs may have demanded increased cognitive resources, which is theoretically related to episodes of mind wandering [24].

Table 6. Standardized coefficients for regressing predicted mind wandering on features in the combined model (*β*)

Features Included in Combined Model	**Standardized Coefficient (β)**
Reading Behavior Features	
Reading Time	**-.750**
Decoupling	**.493**
Textual Features	
Number of Characters	**.139**
Number Words	.099
Referential Cohesion	**-.139**
FKGL	**.239**

Notes. Bold = significant at $p < .05$; FKGL = Flesch Kincaid Grade Level.

Referential Cohesion was also negatively related to predicted mind wandering. This relationship is theoretically plausible, as

breakdowns in *Referential Cohesion* are indicative of increased difficulty [20]. Indeed, difficulty has been found to be related to mind wandering during reading [14, 27].

None of the Context features were included in the combined model. This was an unexpected result, since time on task has previous been correlated to mind wandering [40] and the previous detectors of mind wandering have utilized context features [5, 6]. It is possible that one of the Context Features, *Current Difficulty*, may not have been useful in the combined model due, in part, to the fact that the textual features were essentially more sensitive measures of difficulty. For example, FKGL and Referential Cohesion may be more sensitive measures of *Current Difficulty*.

4.3 Predictive Validity

In order to establish predictive validity for the detector, we ascertained if *predicted* mind wandering relates to learning similar to actual (self-reported) mind wandering rates? Based on previous research, we expect a negative relationship between actual mind wandering and learning [11, 32, 39]. To address this question, posttest performance was first correlated with *actual* rates of mind wandering (i.e., responses to the thought probes). Participants' posttest performance was calculated as the proportion of correct responses for the surface- and inference-level questions separately. The variables were standardized across the two datasets to account for any differences in populations. Indeed, *actual* mind wandering was negatively related to both surface (Spearman's *rho* = -.338, $p < .001$) and inference level (*rho* = -.288, $p < .001$) comprehension on the posttest.

To establish the predictive validity of the detector, we ascertained if *predicted* mind wandering was related to posttest performance similar to actual mind wandering. *Predicted* mind wandering rates (from the combined detector) was negatively correlated with surface level (*rho* = -.294, $p < .001$) as well as inference level performance on the posttest *(rho* = -.193, $p = .008$). The negative correlations with both types of posttest performance gives us some confidence in our model's predictive validity, since predicted mind wandering shows similar relationships with learning as actual self-reported mind wandering. This finding is notable since the model predicted mind wandering correctly around 20% above chance (kappa = .207), yet *predicted* mind wandering related almost as well to posttest scores as *actual* rates of mind wandering.

5. GENERAL DISCUSSION

Mind wandering is a ubiquitous phenomenon that is negatively related to learning [11, 32, 39]. Mind wandering can have a detrimental impact on comprehension when pieces of information are not accurately integrated into a learner's mental model of the instructional texts. Over time, information missed during episodes of mind wandering can accumulate, leaving deficits in the learner's overall understanding of a text. The development of attention-aware systems may provide opportunities to restore learners' attention in real-time to facilitate learning. However, we must first be able to detect mind wandering in order to respond to its occurrence.

We attempted to address this issue by developing a participant-independent detector of mind wandering through analyzing log files and textual characteristics collected during an online reading task. Two diverse datasets were used to ensure further generalizability. The detector was able to accurately classify mind wandering 20% above chance (kappa = .207; *AUC* = .609). Given that mind wandering is an elusive internal state of attention and we used completely sensor-free data, modest classification accuracies are to be expected. Additionally, the classification accuracy found in this study (63%) is only slightly lower than those reported for previous detectors built using sensor-based approaches including eye gaze and physiology (See Table 1; [5, 6].

Three types of features were used to build the mind wandering detector: (1) reading behaviors, (2) textual features, and (3) context features. An independent model was built for each subtype of features, which allowed us to better understand how the subtypes of feature perform independently. Each set of features was able to correctly classify mind wandering independently at levels above chance, though performance varied across models. None of these models outperformed the combined model, so we conclude that combining different types of features was optimal in the current detector. Thus, future research may consider using one or more of these subtypes of features, as they are relatively easy to extract from log files.

Many of the features were included based on previous psychological and educational research on mind wandering. The relationships between the features and predicted rates of mind wandering were revealing in a number of ways. For example, a negative relationship between Referential Cohesion and predicted mind wandering directly supports the situation model view of text comprehension [14, 35]. This view posits that reading involves the construction of a *situation model*, which is a constantly-updated mental representation of a text's meaning [18, 43]. Situation models are harder to construct during difficult texts due to inconsistencies or lack of cohesion. Poorly constructed situation models consume fewer attentional resources, leaving extra resources available for off-task thoughts. Therefore, this theory would predict a negative relationship between mind wandering Referential Cohesion, which is what we find.

Response times as well as reading time information have been utilized in previous detectors of off-task states like disengagement [4, 7, 8]. Thus, it is not surprising that both reading time behavior features were related to predicted mind wandering. A negative relationship with Reading Time indicates that shorter reading times were indicative of increased mind wandering predictions. It is also worth noting that Decoupling, which is derived from a theoretically-supported relationship between reading time and text complexity, was positively related to predicting mind wandering. Indeed, these relationships suggest features based on reading times may be used a behavioral indices of attention during reading.

Our detector also showed some evidence for predictive validity. Predicted mind wandering was negatively related to posttest performance, similar to actual mind wandering. Future work should explore other avenues of establishing validity using other online measures of engagement and comprehension. Similar to [15], another method of validation would be to trigger thought probes on the pages where mind wandering is predicted in real-time. We could then evaluate responses to the predicted episodes of mind wandering in order to determine how accurate the model performs in a real-time detection setting.

It is important to note that these models are not without limitations. First, these models were built in the context of an instructional reading task, which may not generalize to other learning environments. Second, although two independent datasets were used, our results cannot currently be generalized beyond the current sample. Third, although self-reports of mind wandering using a thought-probe method have been validated in previous studies [35, 36], they depend on participants accurate

and honest responses. Additionally, given the internal nature of mind wandering, external coders are not a viable option. Therefore, future work may consider using a different method of probing, where participants might self-monitor and report instances of mind wandering at any point during reading [31] (as opposed to only at times when thought-probes occur). Finally, there is no known research establishing a way to determine the onset of mind wandering in real-time [37]. Thus, while detectors to date are able to predict instances of self-reported mind wandering (which is inherently realized), no method has been established to indicate how long the episode lasts or when it began.

Future work may include attempts to improve these models using additional features. For example, additional sensor-free features, such as trait-based features like prior knowledge and interest might further improve prediction rates. In addition, combining features developed here with previous detectors of mind wandering may also improve prediction rates (e.g., eye gaze). It is possible that combining multiple channels of data may have an additive effect to improve prediction rates.

In summary, this paper provides some initial evidence for a sensor-free detector of mind wandering during online instructional reading. A sensor-free detector of mind wandering may open up new avenues for interventions and instructional designs in order to facilitate attention. Previous detectors for disengagement behaviors, such as gaming the system and Gaze Tutor, have been used in the design of interventions, such as reintroducing the content that is missed due to gaming [2] and providing engaging dialogue to redirect students' attention [12]. The detector presented in this paper is an initial step for interventions that can occur when the mind wanders away from the current task. We believe further development of these types of models is promising for creating an attention-aware system that can respond in real-time.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF; DRL 1235958). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

[1] Arroyo, I. et al. 2007. Repairing disengagement with non-invasive interventions. *AIED* (2007), 195–202.

[2] Baker, R.S.J. et al. 2006. Adapting to when students game an intelligent tutoring system. *Intelligent Tutoring Systems* (2006), 392–401.

[3] Baker, R.S.J. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), 1059–1068.

[4] Beck, J.E. 2004. Using response times to model student disengagement. *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments* (2004), 13–20.

[5] Bixler, R. and D'Mello, S. 2014. Toward Fully Automated Person-Independent Detection of Mind Wandering. *User Modeling, Adaptation, and Personalization*. Springer. 37–48.

[6] Blanchard, N. et al. 2014. Automated Physiological-Based Detection of Mind Wandering during Learning. *Intelligent Tutoring Systems* (2014), 55–60.

[7] Cocea, M. and Weibelzahl, S. 2006. Can Log Files Analysis Estimate LearnersLevel of Motivation?. *LWA* (2006), 32–35.

[8] Cocea, M. and Weibelzahl, S. 2007. Cross-system validation of engagement prediction from log files. *Creating New Learning Experiences on a Global Scale*. Springer. 14–25.

[9] Cocea, M. and Weibelzahl, S. 2011. Disengagement Detection in Online Learning: Validation Studies and Perspectives. *IEEE Transactions on Learning Technologies*. 4, 2 (Apr. 2011), 114–124.

[10] Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*. 20, 1 (1960), 37–46.

[11] Dixon, P. and Bortolussi, M. 2013. Construction, integration, and mind wandering in reading. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*. 67, 1 (2013), 1.

[12] D'Mello, S. et al. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*. 70, 5 (2012), 377–398.

[13] Drummond, J. and Litman, D. 2010. In the zone: Towards detecting student zoning out using supervised machine learning. *Intelligent Tutoring Systems* (2010), 306–308.

[14] Feng, S. et al. 2013. Mind wandering while reading easy and difficult texts. *Psychonomic bulletin & review*. (2013), 1–7.

[15] Franklin, M.S. et al. 2011. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*. 18, 5 (Oct. 2011), 992–997.

[16] Franklin, M.S. et al. 2013. Thinking one thing, saying another: The behavioral correlates of mind-wandering while reading aloud. *Psychonomic Bulletin & Review*. (Jun. 2013).

[17] Graesser, A.C. et al. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*. 36, 2 (2004), 193–202.

[18] Graesser, A.C. et al. 2011. Coh-Metrix Providing Multilevel Analyses of Text Characteristics. *Educational Researcher*. 40, 5 (2011), 223–234.

[19] Graesser, A.C. et al. 2010. What is a good question? *Bringing reading research to life*. Guilford Press. 112–141.

[20] Graesser, A.C. and McNamara, D.S. 2011. Computational analyses of multilevel discourse comprehension. *Topics in Cognitive Science*. 3, 2 (2011), 371–398.

[21] Hanley, J.A. and McNeil, B.J. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 143, 1 (1982), 29–36.

[22] Klare, G.R. 1974. Assessing Readability. *Reading Research Quarterly*. 10, 1 (Jan. 1974), 62–102.

[23] Mason, W. and Suri, S. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*. 44, 1 (2012), 1–23.

[24] McVay, J.C. and Kane, M.J. 2010. Does mind wandering reflect executive function or executive failure? Comment on Smallwood and Schooler (2006) and Watkins (2008). (2010).

[25] Millis, K. et al. 2011. Operation ARIES!: A serious game for teaching scientific inquiry. *Serious games and edutainment applications*. (2011), 169–195.

[26] Mills, C. et al. 2014. To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. *Intelligent Tutoring Systems* (2014), 19–28.

[27] Mills, C. et al. 2013. What Makes Learning Fun? Exploring the Influence of Choice and Difficulty on Mind Wandering and Engagement during Learning. *Artificial Intelligence in Education* (2013), 71–80.

[28] Rand, D.G. 2012. The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments. *Journal of theoretical biology*. 299, (2012), 172–179.

[29] Reichle, E.D. et al. 2010. Eye Movements During Mindless Reading. *Psychological Science*. 21, 9 (Sep. 2010), 1300–1310.

[30] Risko, E.F. et al. 2012. Everyday attention: variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology*. 26, 2 (2012), 234–242.

[31] Schooler, J.W. et al. 2011. Meta-awareness, perceptual decoupling and the wandering mind. *Trends in cognitive sciences*. 15, 7 (2011), 319–326.

[32] Smallwood, J. et al. 2007. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review*. 14, 2 (2007), 230–236.

[33] Smallwood, J. 2011. Mind-wandering while reading: Attentional decoupling, mindless reading and the cascade model of inattention. *Language and Linguistics Compass*. 5, 2 (2011), 63–77.

[34] Smallwood, J. et al. 2009. Shifting moods, wandering minds: negative moods lead the mind to wander. *Emotion*. 9, 2 (2009), 271.

[35] Smallwood, J. et al. 2008. When attention matters: The curious incident of the wandering mind. *Memory & Cognition*. 36, 6 (2008), 1144–1150.

[36] Smallwood, J. and Schooler, J.W. 2006. The restless mind. *Psychological bulletin*. 132, 6 (2006), 946.

[37] Smallwood, J. and Schooler, J.W. 2015. The science of mind wandering: empirically navigating the stream of consciousness. *Annual review of psychology*. 66, (2015), 487–518.

[38] Szpunar, K.K. et al. 2013. Interpolated memory tests reduce mind wandering and improve learning of online lectures. *Proceedings of the National Academy of Sciences*. 110, 16 (2013), 6313–6317.

[39] Szpunar, K.K. et al. 2013. Mind wandering and education: from the classroom to online learning. *Frontiers in psychology*. 4, (2013).

[40] Thomson, D.R. et al. 2014. On the link between mind wandering and task performance over time. *Consciousness and cognition*. 27, (2014), 14–26.

[41] Vega, B. et al. 2013. Reading into the Text: Investigating the Influence of Text Complexity on Cognitive Engagement. *Proceedings of the 6th international conference on educational data mining* (2013), 296–299.

[42] Wixon, M. et al. 2012. WTF? detecting students who are conducting inquiry without thinking fastidiously. *User Modeling, Adaptation, and Personalization*. Springer. 286–296.

[43] Zwaan, R.A. and Radvansky, G.A. 1998. Situation models in language comprehension and memory. *Psychological bulletin*. 123, 2 (1998), 162.

A Comparison of Video-based and Interaction-based Affect Detectors in Physics Playground

Shiming Kai[1], Luc Paquette[1], Ryan S. Baker[1], Nigel Bosch[2], Sidney D'Mello[2], Jaclyn Ocumpaugh[1], Valerie Shute[3], Matthew Ventura[3]

[1]Teachers College Columbia University, 525 W 120th St. New York, NY 10027
[2]University of Notre Dame, 384 Fitzpatrick Hall, Notre Dame, IN 46556
[3]Florida State University, 3205G Stone Building, 1114 West Call Street, Tallahassee, FL 32306
{smk2184, paquette}@tc.columbia.edu, baker2@exchange.tc.columbia.edu, jo2424@tc.columbia.edu, {pbosch, sdmello}@nd.edu, {vshute, mventura}@fsu.edu

ABSTRACT

Increased attention to the relationships between affect and learning has led to the development of machine-learned models that are able to identify students' affective states in computerized learning environments. Data for these affect detectors have been collected from multiple modalities including physical sensors, dialogue logs, and logs of students' interactions with the learning environment. While researchers have successfully developed detectors based on each of these sources, little work has been done to compare the performance of these detectors. In this paper, we address this issue by comparing interaction-based and video-based affect detectors for a physics game called Physics Playground. Specifically, we report on the development and detection accuracy of two suites of affect and behavioral detectors. The first suite of detectors applies facial expression recognition to video data collected with webcams, while the second focuses on students' interactions with the game as recorded in log-files. Ground–truth affect and behavior annotations for both face- and interaction-based detectors were obtained via live field observations during game-play. We first compare the performance of these detectors in predicting students' affective states and off-task behaviors, and then proceed to outline the strengths and weakness of each approach.

Keywords

Video-based detectors, interaction-based detectors, affect, behavior, Physics Playground

1. INTRODUCTION

The development of models that can automatically detect student affect now constitutes a considerable body of research [12,31], particularly in computerized learning contexts [1,34,35], where researchers have successfully built affect-sensitive learning systems that aim to significantly enhance learning outcomes [4,21,30]. In general, researchers attempting to develop affect detectors have developed systems falling into two categories: interaction-based detectors [9] and physical sensor-based detectors [12]. Many successful efforts to detect student affect in intelligent tutoring systems have used visual, audio or physiological sensors, such as webcams, pressure sensitive seat or back pads, and pressure-sensing keyboards and mice [3,28,37,41].

The development of sensor-based detectors has progressed significantly over the last decade, but one limitation to this research is that much of it has taken place in laboratory conditions, which may not generalize well to real-world settings [9]. While efforts are being made to address this issue [4], there are often serious obstacles to using sensors in regular classrooms. For example, sensor equipment may be bulky or otherwise obtrusive, distracting students from their primary tasks (learning); sensors may also be expensive and prone to malfunction, making large-scale implementation impractical, particularly for schools that are already financially strained. On the other hand, because physical sensors are external to specific learning systems, their use in affect detection creates the opportunity for them to be applied to entirely new learning systems, though this possibility has yet to be empirically tested.

Interaction-based detection [9] has also improved over the last decade. Unlike sensor-based detectors, which rely upon the physical reactions of the student, these detectors infer affective states from students' interactions with computerized learning systems [5,7,9,14,29,30]. The fact that interaction-based affect detectors rely on student interactions makes it possible for them to run in the background in real time at no extra cost to a school that is using the learning system. Their unobtrusive and cost-efficient nature also makes it feasible to apply interaction-based detectors at scale, leading to a growing field of research regarding discovery with models [8]. For example, interaction-based affect detection has been useful in predicting student long-term outcomes, including standardized exam scores [30] and college attendance [36]. Basing affect detection on student interactions with the system, however, give rise to issues with generalizing such detectors across populations [26] and learning systems. Because interaction-based detectors are highly dependent on the computation of features that captures the student's interactions with the specific learning platform, the type of features generated is contingent on the learning system itself, making it difficult to apply the same sets of features across different systems.

It has become clear that each modeling approach has its own utility; researchers have thus begun to speculate on effectiveness across the various approaches and the possible applications of multimodal detectors. However, the body of research that addresses this question is currently quite limited. Arroyo and colleagues [4] applied sensor-based detectors in a classroom setting, and compared performances between interaction-only detectors and detectors using both interaction and sensor data, in predicting student affect. They found that the inclusion of sensor data in the detectors improved performance and accuracy in

identifying student affect. However, a direct comparison between the two types of detectors was not made. Furthermore, the sample size tested was relatively small (26-30 instances depending on model), and the data was not cross-validated. Comparisons between types of detectors were made in D'Mello and Graesser's study [18], which compared interaction, sensor and face-based detectors in an automated tutor. They found face-based detectors to perform better than interaction and posture-based detectors at predicting spontaneous affective states. However, the study was conducted in a controlled laboratory setting, and the facial features recorded were manually annotated.

In this paper, we build detectors of student affect in classroom settings, using both sensor-based and interaction-based approaches. For feasibility of scaling, we limit physical sensors to webcams. For feasibility of comparison, the two types of detectors are built in comparable fashions, using the same ground truth data obtained from field observations that were conducted during the study. We conduct this comparison in the context of 8th and 9th grade students playing an educational game, Physics Playground, in the Southeastern United States. Different approaches were used to build each suite of detectors in order to capitalize on the affordances of each modality. However, the methods and metrics to establish accuracy were held constant in order to render the comparison meaningful.

2. PHYSICS PLAYGROUND

Physics Playground (formerly, Newton's Playground, see [39]) is a 2-dimensional physics game where students apply various Newtonian principles as they create and guide a ball to a red balloon placed on screen [38]. It offers an exploratory and open-ended game-like interface that allows students to move at their own pace. Thus, Physics Playground encourages conceptual learning of the relevant physics concepts through experimentation and exploration. All objects in the game obey the basic laws of physics, (i.e., gravity and Newton's basic laws of motion).

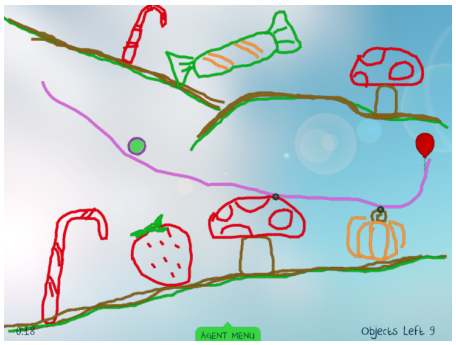


Figure 1: Screenshot of Physics Playground

Students can choose to enter one of seven different playgrounds, and then play any of the 10 or so levels within that playground. Each level consists of various obstacles scattered around the space, as well as a balloon positioned at different locations within the space (see Figure 1). Students can nudge the ball left and right, but will need to create simple machines (called "agents of force and motion" in the game) on-screen in order to solve the problems presented in the playgrounds. There are four possible agents that may be created: ramps, pendulums, levers and springboards. Students can also create fixed points along a line drawing to create pivots for the agents they create. Students use the mouse to draw agents that come to life after being drawn, and use them to propel the ball to the red balloon. Students control the weight and density of objects through their drawings, making an object denser, for example, by filling it with more lines.

Each level allows multiple solutions, encouraging students to experiment with various methods to achieve the goal and guide the ball towards the balloon. Trophies are awarded both for achieving the goal objective and for solutions deemed particularly elegant or creative, encouraging students to attempt each playground more than once. This unstructured game-like environment provides us with a rich setting in which to examine the patterns of students' affect and behavior as they interact with the game platform.

3. DATA COLLECTION

Students in the 8th and 9th grade were selected due to the alignment of the curriculum in Physics Playground to the state standards at those grade levels. The student sample consisted of 137 students (57 male, 80 female) who were enrolled in a public school in the Southeastern U.S. Each group of about 20 students used Physics Playground during 55-minute class periods over the course of four days.

An online physics pretest (administered at the start of day 1) and posttest (administered at the end of day 4), measured student knowledge and skills related to Newtonian physics. In this paper, our focus is on data collected during days 2 and 3, during which time students were participating in two full sessions of game play.

The study was conducted in a computer-enabled classroom with 30 desktop computers. Inexpensive webcams ($30 each) were affixed at the top of each computer monitor. At the beginning of each session, the webcam software displayed an interface that allowed students to position their faces in the center of the camera's view by adjusting the camera angle up or down. This process was guided by on-screen instructions and verbal instructions from the experimenters, who were available to answer any additional questions and to troubleshoot any problems.

3.1 Field Observations

Students were observed by two BROMP-certified observers while using the Physics Playground software. The Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP 2.0) is a momentary time sampling system that has been used to study behavioral and affective indicators of student engagement in a number of learning environments [9]. BROMP coders observe each student individually, in a predetermined order. They record only the first predominant behavior and affect that the student displays, but they have up to 20 seconds to determine what that might be.

In this study, BROMP coding was done by the 6th author and the 4th author. The 6th author, a co-developer of BROMP, has been validated to achieve acceptable inter-rater reliability (kappa >= 0.60) with over a dozen other BROMP-certified coders. The 4th author achieved sufficient inter-rater reliability (kappa >= 0.60) with the 6th author on the first day of this study.

The coding process was implemented using the Human Affect Recording Tool (HART) application for Android devices [6], which enforces the protocol while facilitating data collection. The study used coding schema that had previously been used in several other studies of student engagement [e.g. 17], and included *boredom, confusion, engaged concentration,* and *frustration* (affective states) as well as *on task, on-task conversation,* and *off-task* (behavioral states). Consistent with previous BROMP research, "*?*" was recorded when a student could not be coded, when an observer was unable to identify the

student's behavior or affective state, or when the affect/behavior of the student was clearly a construct outside of the coding scheme (such as *anger*).

Modifications to the affective coding scheme were made on the third day of the study, with the addition of *delight* and *dejection*. *Delight* was defined as a state of strong positive affect, often indicated by broad smiling or a student bouncing in his/her chair. This affective state had been coded in previous studies (see [9]), and was used to construct detectors. *Dejection*, defined as a state of being saddened, distressed, or embarrassed by failure [9], is likely the affect that corresponds with the experience of *stuck* [11,20]. Because it had not been coded in previous research, and because it was still quite rare in Physics Playground, it was not modeled for this study.

3.2 Affect and Behavior Incidence

An initial number of 2,374 observations were made across all 137 students during the course of the study, culminating in 17.3 observations made per student across the second and third days of the study Only affect observations on the second and third days were used in the construction of the detectors, since the first and last days mostly consisted of pretests and posttests. Other observations were dropped as a result of two students who switched computers halfway through data collection, resulting in each student being logged under the other student's ID for part of the study. The remaining 2,087 observations recorded during the second and third days were used in the construction of both detectors. An additional 214 were removed prior to the construction of the interaction-based detectors and 863 were removed prior to the construction of the video-based detectors. Because the criteria for these exclusions were methodologically based, further details are provided in the sections describing the construction of each detector.

Within the field observations, the most common affective state observed was *engaged concentration* with 1293 instances (62.0%), followed by *frustration* with 235 instances (11.3%). *Boredom and confusion* were far less frequent despite being observed across both second and third days of observation: 66 instances (3.2%) for *boredom* and 38 instances (1.8%) for *confusion*. *Delight* was only coded on the third day, and was also rare (45 instances), but it still comprised 2.2% of the total observations.

The frequency of off-task behavior observations was 4.0% (84 instances), which was unusually low compared to prior classroom research in the USA using the same method with other educational technologies [27,33]. On-task conversation was seen 18.6% of the time (388 instances).

4. INTERACTION-BASED DETECTORS

To create interaction affect detectors, BROMP affect observations were synchronized to the log files of student interactions with the software. Features were then generated and a 10-fold student-level cross validation process was applied for machine learning, using five classification algorithms.

4.1 Feature Engineering

The feature engineering process for this study was based largely on previous research on student engagement, learning, and persistence. The initial set of features comprised 76 gameplay attributes that potentially contain evidence for specific affective states and behavior. Some attributes included:

- The total number of springboard structures created in a level
- The total number of freeform objects drawn in a level
- The amount of time between start to end of a level
- The average number of gold and silver trophies obtained in a level
- The number of stacking events (gaming behavior) in a level

Features created may be grouped into two broad categories. Time-based features focus on the amount of time elapsed between specific student actions, such as starting and pausing a level, as well as the time it takes for a variety of events to occur within each playground level. Other features take into account the number of specific objects drawn or actions and events occurring during gameplay, given various conditions.

Missing values were present at certain points in the dataset when a particular interaction was not logged. For example, a feature specifying the amount of time between the student beginning a level and his/her first restart of the level, would contain a missing value if the student manages to complete a level without having to restart it. A variety of data imputation approaches were used in these situations to fill in the missing values so that we could retain the full sample size. We used single, average and zero imputation methods to fill in the missing data, and ran the new datasets through the machine learning process to identify the best data imputation strategy for each affect detector. Zero imputations were performed where the missing values were replaced by the value 0, while average data imputations took place when the average value for the particular feature was computed, and the missing values replaced by this average value. In single data imputation, we used RapidMiner to build an M5' model [32], a tree-based decision model, to predict the values for each feature, and applied the model to compute a prediction of the missing value. We also ran the original dataset without any imputation through any of the classification algorithms that allowed it.

Of the 2087 BROMP field observations that were collected, 214 instances were removed as most of these instances corresponded to times when the student was inactive. Additional instances were removed where the observer recorded a *?*, the code used when BROMP observers cannot identify a specific affect or behavior or when students are not at their workstation. In total, 171 instances of affect and 63 instances of behavior were coded as ?. As a result, these instances did not contribute to the building of the respective affect and behavior detectors.

4.2 Machine Learning

Data collection was followed by a multi-step process to develop interaction-based detectors of each affect. A two-class approach was used for each affective state, where that affective state was discriminated from all others. For example, engaged concentration was discriminated from all frustrated, bored, delighted, and confused instances combined (referred to as "all other"). Behaviors were grouped into two classes: 1) off task, and 2) both on task behaviors and on task conversation related to the game.

4.2.1 Resampling of Data

Because observations of several of the constructs included in this study were infrequent, (< 5.0% of the total number of observations), there were large class imbalances in our data distributions. To correct for this, we used the *cloning* method for resampling, generating copies of respective positive affect on the training data, in order to make class frequency more balanced for detector development.

4.2.2 Feature Selection and Cross-Validation

Correlation-based filtering was used to remove features that had very low correlation with the predicted affect and behavior constructs (correlation coefficient > 0.04) from the initial feature set. Feature selection for each detector was then conducted using forward selection.

Detectors for each construct were built in the RapidMiner 5.3 data-mining software, using common classification algorithms that have been previously shown to be successful in building affect detectors: JRip, J48 decision trees, KStar, Naïve-Bayes, step and logistic regression. Models were validated using 10-fold student-level batch cross-validation. The performance metric of A' was computed on the original, non-resampled, datasets.

4.3 Selected Features

From the forward selection process, a combination of features was selected in each of the affect and behavior detectors that provide some insight into the type of student interactions that predict the particular affective state or behavior.

The features for *boredom* involve a student spending more time between actions on average. A bored student would also expend less effort to guide the ball object to move in the right direction, as indicated by fewer nudges made on the ball object to move it, and more ball objects being lost from the screen.

The features that predict *confusion* are characterized by a student spending more time before his/her first nudge to make the ball object move, and drawing fewer objects in a playground level. A student who is confused may not have known how to draw and move the ball object towards the balloon, thus spending a long time within a certain level and resulting in a lower number of levels attempted in total.

From the features selected, *delight* appears to ensue from some indicator of success, such as a student who is able to achieve a silver trophy earlier on during gameplay, and who completes more levels in total. We can also portray the student who experiences *delight* as someone who was able to achieve the objective without having to make multiple attempts to draw the relevant simple machines (such as springboards and pendulums).

The features for *engaged concentration* would describe a student who is able to complete a level in fewer attempts but erases the ball object more often during each attempt, indicating that the student was putting in more effort to refine his/her strategies within a single attempt at the level. *Engaged concentration* would also depict a student who has experienced success during gameplay and achieved a silver trophy in a shorter than average time, perhaps because of his/her focused efforts during each attempt.

Table 1. Features in the final interaction-based detectors of each construct

Affect/ Behavior	Selected features
Boredom	Time between actions within a level
	Total number of objects that were "lost" (i.e. Moved off the screen)
	Total number of nudges made on the ball object to move it
Confusion	Amount of time spent before the ball object was nudged to move
	Total number of levels attempted
	Total number of objects drawn within the level
Delight	Number of silver trophies achieved
	Consecutive number of pendulums and springboards created
	Total number of levels attempted
	Total number of levels completed successfully
Engaged Concentration	Total number of silver trophies achieved in under the average time
	Total number of level re-starts within a playground
	Total number of times a ball object was erased consecutively
Frustration	Total number of silver trophies achieved in under the average time
	Total number of level re-starts within a playground
	Total number of levels completed successfully
	Total number of levels attempted
Off-task Behavior	Time spent in between each student action
	Total number of pauses made within a level
	Total number of times a student quits a level without completing the objective and obtaining a trophy

Unlike *engaged concentration*, a student who experiences *frustration* failed to achieve the objective and achieved fewer silver trophies within the average time taken. Student *frustration*, as seen in the features, would also result in the student having to make more attempts at a level due to repeated failure, thus resulting in fewer levels attempted in total.

Lastly, behavior that is *off-task* involves a student who spends more time pausing the level or between actions as a whole. It is also apparent in a student who draws fewer objects and quits more levels without completing them, implying that he or she did not put in much effort to complete the playground levels.

5. VIDEO-BASED DETECTORS

The video-based detectors have been reported in a recent publication [10]. In the interest of completeness, the main approach is re-presented here. There are also small differences in the results reported here due to a different validation approach that was used to make meaningful comparisons with interaction-based detectors.

Video-based affect detectors were constructed using FACET (no longer available as standalone software), a commercialized version of the Computer Expression Recognition Toolbox (CERT) software [25]. FACET is a computer vision tool used to automatically detect Action Units (AUs), which are labels for specific facial muscle activations (e.g. lowered brow). AUs provide a small set of features for use in affect detection efforts. A large database of AU-labeled data can be used to train AU

detectors, which can then be applied to new data to generate AU labels.

5.1 Feature Engineering

FACET provides estimates of the likelihood estimates for the presence of nineteen AUs as well as head pose (orientation) and position information detected from video. Data from FACET was temporally aligned with affect observations in small windows. We tested five different window sizes (3, 6, 9, 12, and 20 seconds) for creation of features. Features were created by aggregating values obtained from FACET (AUs, orientation and position of the face) in a window of time leading up to each observation using maximum, median, and standard deviation. For example, with a six-second window we created three features from the AU4 channel (brow lowered) by taking the maximum, median, and standard deviation of AU4 likelihood within the six seconds leading up to an affect observation. In all there were 78 facial features.

We used features computed from gross body movement present in the videos as well. Body movement was calculated by measuring the proportion of pixels in each video frame that differed from a continuously updated estimate of the background image generated from the four previous frames. Previous work has shown that features derived using this technique correlate with relevant affective states including boredom, confusion, and frustration [17]. We created three body movement features using the maximum, median, and standard deviation of the proportion of different pixels within the window of time leading up to an observation, similar to the method used to create FACET features.

Of the initial 2087 instances available for us to train our video-based detectors on, about a quarter (25%) were discarded because FACET was not able to register the face and thus could not estimate the presence of AUs and computation of features. Poor lighting, extreme head pose or position, occlusions from hand-to-face gestures, and rapid movements can all cause face registration errors; these issues were not uncommon due to the game-like nature of the software and the active behaviors of the young students in this study. We also removed 9% of instances because the window of time leading up to the observation contained less than one second (13 frames) of data in which the face could be detected, culminating in 1224 instances where we had sufficient video data to train our affect models on.

5.2 Machine Learning

We also built separate detectors for each affective state similar to the interaction-based detectors. Building individual detectors for each state allows the parameters (e.g., window size, features used) to be optimized for that particular affective state.

5.2.1 Resampling of Data

Like the interaction-based detectors, there were large class imbalances in the affective and behavior distributions. Two sampling techniques, different from the one used in the building of interaction-based detectors, were used on the training data to compensate for this imbalance. These two techniques included downsampling (removal of random instances from the majority class) and synthetic oversampling (with SMOTE; [13]) to create equal class sizes. SMOTE creates synthetic training data by interpolating feature values between an instance and randomly chosen nearest neighbors. The distributions in the testing data were not changed, to preserve the validity of the results.

5.2.2 Feature Selection and Cross-Validation

We used tolerance analysis to eliminate features with high multicollinearity (variance inflation factor > 5) [2]) for video-based detectors. Feature selection was then used to obtain a more diagnostic set of features for classification. RELIEF-F [24] was run on the training data in order to rank features. A proportion of the highest ranked features were then used in the models (.1, .2, .3, .4, .5, and .75 proportions were tested). A detailed analysis or table of the features selected for the video-based detectors is not included because of the large number of features utilized by these detectors.

We then built classification models using 14 different classifiers including support vector machines, C4.5 trees, Bayesian classifiers, and others in the Waikato Environment for Knowledge Analysis (WEKA), a machine learning tool [23].

6. RESULTS

We evaluated the extent to which the detectors for each construct are able to identify their respective affect. Both detectors were evaluated using a 10-fold student-level batch cross-validation. In this process, students in the training dataset are randomly divided into ten groups of approximately equal size. A detector is built using data from all possible combinations of 9 out of the overall 10 groups, and finally tested on the last group. Cross-validation at this level increases the confidence that the affect and behavior

Table 2. A' performance values for affect and behavior using video-based and interaction-based detectors

Affect/Behavior Construct	Interaction-Based Detectors				Video-Based Detectors		
	Classifier	Data Imputation Scheme	A'	No. Instances	Classifier	A'	No. Instances
Boredom	Logistic regression	Zero	0.629	1732	Classification via Clustering	0.617	1305
Confusion	Step regression	Average	0.588	1732	Bayes Net	0.622	1293
Delight	Logistic regression	None	0.679	1732	Updateable Naïve Bayes	0.860	1003
Engaged Concentration	Naïve Bayes	Zero	0.586	1732	Bayes Net	0.658	1228
Frustration	Logistic regression	Average	0.559	1732	Bayes Net	0.632	1132
Off-Task behavior	Step regression	Zero	0.765	1829	Logistic Regression	0.780	1381

detectors will be more accurate for new students. To ensure comparability between the two sets of detectors, the cross-validation process was carried out with the same randomly selected groups of students.

Detector performance was assessed using A' values that were computed as the Wilcoxon statistic [22]. A' is the probability that the given algorithm will correctly identify whether an observation is an example of a specific affective state. A' can be approximated by the Wilcoxon statistic and is equivalent to the area under the Receiver Operating Characteristic (ROC) curve in signal detection theory. A detector with a performance of A' = 0.5 is performing at chance, while a model with a performance of A' = 1.0 is performing with perfect accuracy.

Table 2 shows the performance of the two detector suites. Both interaction-based and video-based detectors' performance over all six affective and behavior constructs was better than chance (A' = 0.50). On average, the interaction-based detectors yielded an A' of 0.634 while the video-based detectors had an average A' of 0.695. This difference can be mainly attributed to the detection of delight, which was much more successful for the video-based detectors. Accuracy of the two detector suites was much more comparable for the other constructs, though the video-based detectors showed some advantages for engaged concentration and frustration, and were higher for 5 of the 6 constructs.

The majority of the video-based detectors performed the best when using the Bayes Net classifier, except for *boredom*, *delight* and *off-task behavior*. In comparison, logistic and step regression composed the classifiers that produced the best performance for most of the interaction-based detectors, with the exception of *engaged concentration*.

7. DISCUSSION

Affect detection is becoming an important component in educational software, which aims to improve student outcomes by dynamically responding to student affect. Affect detectors have been successfully built and implemented via different modalities [3,16,41], and each have their own advantages and disadvantages when implemented in a noisy classroom environment. This study is an extension of previous research conducted on both video-based and interaction-based detectors. Having been mostly built in controlled laboratory settings [12], we now test the performance for video-based detectors within an uncontrolled computer-enabled classroom environment that is more representative of an authentic educational setting. Although interaction-based detectors have been built to some degree of success in whole classroom settings [5,7,29], we now test the performance of these affect detectors in an open-ended and exploratory educational game platform.

In this paper, we compared the performances of six video-based and interaction-based detectors on student affect and behavior in the game-based software. We will discuss the implications of these comparisons in this section, as well as future work.

7.1 Main Findings

The performances of both detectors in the six affects and off-task behavior appear to be at similar levels above chance for five of the constructs, with video-based detectors performing slightly better than interaction-based detectors on the whole, and with video-based detector showing a stronger advantage for delight. Several factors may have help to explain the relative performances.

Performance of video detectors could be influenced by the uncontrolled whole-classroom setting in which video data is collected, where there are higher chances of video data being absent or compromised due to unpredictable student movement. While there were initially 2,087 instances of affect and behavior observed and coded, a moderate proportion of facial data instances were dropped from the final dataset when building the models. There were 44 instances of affect observation that were dropped either because the video was corrupted or incomplete, or because no video was recorded at all. In addition, there were 520 instances where video was recorded, but facial data were not detected for some reason, perhaps because the student had left the workstation, or when the face could not be detected in the video. An additional 211 instances were removed even though facial data was detected, because the facial data recorded was present for less than 1 second, such that no features could be calculated.

For interaction-based detectors, the exploratory and open-ended user-interface [40] constitutes a unique challenge in creating accurate models for student affect and behavior. The open-ended interface included multiple goals and several possible solutions that students could come up with to successfully complete each level. During gameplay, there are also multiple factors that could contribute to a student's failure to complete a level, such as conceptual knowledge as well as implementation of appropriate objects. A student with accurate conceptual knowledge of simple machines and Newtonian physics may still fail the level because of problems implementing the actions needed to guide the ball to the target. On the other hand, a student with misconceptions about the relevant physics topics may nevertheless be able to complete the level successfully through systematic experimentation. The possible combinations of student actions that result in failure or success in a playground level would hence contribute to the lower accuracy of interaction-based detectors on identifying students' affect based on their interactions with the software.

Another issue with the Physics Playground software could be that there are fewer indicators of success per unit of time, as compared to other learning software that have been studied previously, such as the Cognitive Tutors [e.g. 5]. During gameplay, the system is able to recognize when combinations of objects the student draws forms an eligible agent. However, this indicator of success or failure is not apparent to the student until after he or she creates the ball object and applies a relevant force to trigger a simulation. Since students often spend at least several minutes building agents and ball objects, this results in coarser-grained indicators and evaluations of success and failure. This is in comparison to affect detectors created in previous studies for the Cognitive Tutor software, in which there was regular evaluation of each question attempted, thus resulting in more indicators of success over a given time period. The combination of open-endedness and lack of success indicators per unit of time consequently leads to greater difficulty translating the semantics of student-software interactions into accurate affect predictions.

When comparing between the two sets of detectors, physical detectors make direct use of students' facial features and bodily movements captured by webcams and constitute embodied representations of students' affective states. On the other hand, interaction detectors were built based on student actions within the software, which serves as an indirect proxy of the students' actual affective states. These detectors rely, therefore on the degree to which student interactions with the software are influenced (or not) by the affective states they experience. Perhaps not surprisingly, video-based detectors perform somewhat better

in predicting some affective states (e.g., delight, engaged concentration, and frustration). Although the video detectors are limited by missing data, interaction-based detectors can only detect something that causes students to change their behaviors within the software, which can be challenging given the issues arising from the open-ended game platform. Simply put, face-based affect detectors appear to provide more accurate affect estimates but in fewer situations, while interaction-based affect detectors provide less accurate estimates, but are applicable in more situations. The two approaches thus appear to be quite complementary.

7.2 Limitations

In comparing the performances between interaction and video-based detectors, there exist several limitations in ensuring an equivalent set of methods for a fair comparison to be made.

Although both types of detectors were built based on the same ground truth data, varying sets of limitations exist that are unique to each set of detectors. A smaller proportion of instances were retained to build video-based detectors due to missing video data, which may influence performance comparison. Interaction-based detectors, on the other hand, are relatively more sensitive to the type of educational platform it is built upon, as compared to video-based detectors. The type of learning platform thus affects the variety of features that are relevant and useful in building the affect and behavior detectors, which in turn impacts its performance relative to previous work.

For both detectors, the sample size available for some of the affective states was quite limited, which made it necessary to oversample the training data in order to compensate for the class imbalances. However, because each detector was built on different platforms, different methods were used in oversampling the datasets. The need to conduct data imputations was also unique to interaction-based detectors due to the nature of some of the computed features, and not required for video-based detectors. The difference in these methods may in turn affect performance comparison between the two types of detectors.

7.3 Concluding Remarks

Given the various advantages and limitations to each type of detector in accurately predicting student affect, it may be beneficial for affect detection strategies to include a combination of video-based and interaction-based detectors. While video-based detectors provide more direct measures of student affect, practical issues may lead to video data being absent or unusable in detecting affect, simply because there is no facial data available to detect affect in. These situations may be alleviated by the presence of interaction data that are recorded automatically during students' use of the software. On the other hand, video-based facial data would be able to provide support to interaction data and boost the accuracy in which affective states are detected among students. This form of late-fusion or decision-level fusion can also be complemented by early-fusion or feature-level fusion, where features from both modalities are combined prior to classification. Whether this leads to improved accuracy, as routinely documented in the literature on multimodal affect detection [15,16] awaits future work.

8. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) (DRL 1235958) and the Bill & Melinda Gates Foundation. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF or the Bill & Melinda Gates Foundation.

9. REFERENCES

[1] Ai, H., Litman, D.J., Forbes-Riley, K., Rotaru, M., Tetreault, J., and Purandare, A. 2006. Using system and user performance features to improve emotion detection in spoken tutoring dialogs. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 797–800.

[2] Allison, P.D. 1999. *Multiple regression: A primer.* Pine Forge Press.

[3] AlZoubi, O., Calvo, R. a., and Stevens, R.H. 2009. Classification of EEG for affect recognition: An adaptive approach. *Lecture Notes in Computer Science 5866 LNAI*, , 52–61.

[4] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., and Christopherson, R. 2009. Emotion sensors go to school. *Frontiers in Artificial Intelligence and Applications*, 17–24.

[5] Baker, R.D., Gowda, S., and Wixon, M. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proceedings of the 5th International Conference on Educational Data Mining*, 126–133.

[6] Baker, R.D., Gowda, S., Wixon, M., et al. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proceedings of the 5th International Conference on Educational Data Mining*, 126–133.

[7] Baker, R.S., Ocumpaugh, J., Gowda, S.M., Kamarainen, A.M., and Metcalf, S.J. 2014. Extending Log-Based Affect Detection to a Multi-User Virtual Environment for Science. *22nd Conference on User Modeling, Adaptation and Personalization (UMAP 2014)*, 290–300.

[8] Baker, R.S.J.D. and Yacef, K. 2009. The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining 1*, 1, 3–16.

[9] Baker, R.S.; Ocumpaugh, J. 2015. Interaction-Based Affect Detection in Educational Software. In R.A. Calvo, S.K. D'Mello, J. Gratch and A. Kappas, eds., *Handbook of Affective Computing*. Oxford University Press, Oxford, UK, 233–245.

[10] Bosch, N., Mello, S.D., Baker, R., et al. Automatic Detection of Learning - Centered Affective States in the Wild. *In Proceedings of the 2015 International Conference on Intelligent User Interfaces (IUI 2015). New York, NY, USA: ACM.*

[11] Burleson, W. and Picard, R.W. 2004. Affective agents: Sustaining motivation to learn through failure and a state of stuck. *Proceedings of the Workshop on Social and Emotional Intelligence in Learning Environments in conjunction with the seventh International Conference on Intelligent Tutoring Systems (ITS)*.

[12] Calvo, R.A.. and D'Mello, S.K. 2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and their Application to Learning Environments. *IEEE Transactions on Affective Computing 1*, 1, 18–37.

[13] Chawla, N. V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. 2011. SMOTE: synthetic minority over-sampling

technique. *Journal of Artificial Intelligence Research 16*, , 321–357.

[14] D'Mello, S., Jackson, T., Craig, S., et al. 2008. AutoTutor Detects and Responds to Learners Affective and Cognitive States. *Proceedings of the Workshop on Emotional and Cognitive issues in ITS in conjunction with the 9th International Conference on ITS*, 31–43.

[15] D'Mello, S. and Kory, J. A Review and Meta-Analysis of Multimodal Affect Detection. *ACM Computing Surveys*, .

[16] D'Mello, S. and Kory, J. 2012. Consistent but Modest: A Meta-Analysis on Unimodal and Multimodal Affect Detection Accuracies from 30 Studies. *ACM International Conference on Multimodal Interaction*, 31–38.

[17] D'Mello, S. 2011. Dynamical emotions: bodily dynamics of affect during problem solving. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.

[18] D'Mello, S.K. and Graesser, A. 2010. Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modelling and User-Adapted Interaction 20*, 2, 147–187.

[19] D'Mello, S.K. 2013. A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology 105*, 4, 1082–1099.

[20] D'Mello, S.K.; Graesser, A. 2012. Emotions During Learning with AutoTutor. In *Adaptive Technologies for Training and Education*. 169–187.

[21] Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., El Kaliouby, R., and Eydgahi, H. 2008. Viewing student affect and learning through classroom observation and physical sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5091 LNCS*, 29–39.

[22] Hanley, J.A. and Mcneil, B.J. 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology 143*, 29–36.

[23] Holmes, G., Donkin, A., and Witten, I.H. 1994. WEKA: a machine learning workbench. *Proceedings of ANZIIS '94 - Australian New Zealand Intelligent Information Systems Conference*, 357–361.

[24] Kononenko, I. 1994. Estimating attributes: Analysis and extensions of RELIEF. In F. Bergadano and L. De Raedt, eds., *Machine Learning: ECML-94*. Springer, Berlin Heidelberg, 171–182.

[25] Littlewort, G., Whitehill, J., Wu, T., et al. 2011. The Computer Expression Recognition Toolbox (CERT). *International Conference on IEEE*, 298–305.

[26] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., and Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology 45*, 3, 487–501.

[27] Ocumpaugh, J., Baker, R.S.J., Gaudino, S., Labrum, M.J., and Dezendorf, T. 2013. Field Observations of Engagement in Reasoning Mind. *Proceedings of the 16th International Conference on Artificial Intelligence and Education*, 624–627.

[28] Pantic, M., Pantic, M., Rothkrantz, L.J.M., and Rothkrantz, L.J.M. 2003. Toward an Affect-Sensitive Multimodal Human Computer Interaction. *Proceedings of the IEEE 91*, 9, 1370–1390.

[29] Paquette, L., Baker, R.S.J. d., Sao Pedro, M., et al. 2014. Sensor-Free Affect Detection for a Simulation-Based Science Inquiry Learning Environment. *Proceedings of the 12th International Conference on ITS 2014*, 1–10.

[30] Pardos, Z. a., Baker, R.S.J. d., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2014. Affective states and state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *Journal of Learning Analytics 1*, 1, 107–128.

[31] Picard, R.W. 1997. *Affective Computing.* MIT Press, Cambridge, MA.

[32] Quinlan, J.R. 1992. Learning with continuous classes. *Machine Learning 92*, 343–348.

[33] Rodrigo, M., Baker, R., and Rossi, L. 2013. Student Off-Task Behavior in Computer-Based Learning in the Philippines: Comparison to Prior Research in the USA. *Teachers College Record 115*, 10, 1–27.

[34] Rodrigo, M.M.T. and Baker, R.S.J. d. 2009. Coarse-grained detection of student frustration in an introductory programming course. *Proceedings of the fifth International Computing Education Research Workshop - ICER 2009*.

[35] Sabourin, J., Mott, B., and Lester, J. 2011. Modeling Learner Affect with Theoreticallly Grounded Dynamic Bayesian Networks. *Proceedings of the Fourth International Conference on Affective Computing and Intelligent Interaction*, 286–295.

[36] San Pedro, M.O.Z., Baker, R.S.J. d., Bowers, A.J., and Heffernan, N.T. 2013. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. *Proceedings of the 6th International Conference on Educational Data Mining*, 177–184.

[37] Sebe, N., Cohen, I., Gevers, T., and Huang, T.S. 2005. Multimodal Approaches for Emotion Recognition: A Survey. *Proceedings of SPIE – The International Society for Optical Engineering*, 56–67.

[38] Shute, V., Ventura, M., and Kim, Y.J. 2013. Assessment and Learning of Qualitative Physics in Newton ' s Playground Newton ' s Playground. *The Journal of Educational Research 29*, 579–582.

[39] Shute, V. and Ventura, M. 2013. *Measuring and Supporting Learning in Games Stealth Assessment.* MIT Press, Cambridge, MA.

[40] Shute, Valerie; Ventura, Matthew; Kim, Y.J. 2013. Assessment and Learning of Qualitative Physics in Newton's Playground. *Journal of Educational Research 106*, 423–430.

[41] Zeng, Z., Pantic, M., Roisman, G.I., and Huang, T.S. 2009. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 39–58.

Exploring Dynamical Assessments of Affect, Behavior, and Cognition and Math State Test Achievement

Maria Ofelia Z. San Pedro[1], Erica L. Snow[2], Ryan S. Baker[1], Danielle S. McNamara[2], Neil T. Heffernan[3]
[1]Teachers College Columbia University, 525 W 120th St. New York, NY 10027
[2]Arizona State University, Learning Sciences Institute, 1000 S. Forest Mall, Tempe, AZ 85287
[3]Worcester Polytechnic Institute, 100 Institute Rd. Worcester, MA 01609

mzs2106@tc.columbia.edu, Erica.L.Snow@asu.edu, baker2@exchange.tc.columbia.edu, Danielle.McNamara@asu.edu, nth@wpi.edu

ABSTRACT
There is increasing evidence that fine-grained aspects of student performance and interaction within educational software are predictive of long-term learning. Machine learning models have been used to provide assessments of affect, behavior, and cognition based on analyses of system log data, estimating the probability of a student's particular affective state, behavior, and knowledge (cognition). These measures have (in aggregate) successfully predicted outcomes such as performance on standardized exams. In this paper, we employ a different approach of relating interaction patterns to learning outcomes, using dynamical methods that assess patterns of fine-grained measures of affect, behavior, and knowledge as they occur across time. We use Hurst exponents and Entropy scores computed from assessments of affect, behavior, performance, and knowledge acquired from 1,376 middle school students who used a math tutoring system (ASSISTments), and analyze the relations of these dynamical measures to the students' end-of-year state test (MCAS) performance. Our results show that fine-grained changes in affect, behavior, and knowledge are significantly related to and predictive of their eventual MCAS performance, providing a new lens on the dynamic and nuanced nature of student interaction within online learning platforms and how it affects achievement.

Keywords
Affect Detection, Knowledge Modeling, Educational Data Mining, Hurst, Entropy

1. INTRODUCTION
The increasing deployment of educational software in classrooms has provided new opportunities for studying a broad range of student modeling constructs. The ability of these systems to log student interaction in fine-grained detail has led to the development of automated detectors or models of student learning and engagement [1, 4, 5, 6, 10]. It has been demonstrated through *discovery with models* analyses [20] that detector assessments of engagement and learning can be used to predict long-term student outcomes such as performance in end-of-year standardized exams [24], college enrollment [31] and college major choice [33], even several years after the student engages in online learning. The fine-grained measures of learning and engagement at the action level are then aggregated at the student-level in forming a training dataset for the prediction of learning outcomes. However, these assessments often use simple aggregation methods such as student-level averages, whereas it is known that there are complex patterns in how affect develops over time (e.g. [14]). Hence these simple methods of aggregation may miss fine-grained and nuanced patterns in affect or behaviors that manifest across time.

Indeed, research has also shown that students' learning behaviors are complex and dynamic in nature [19]. Recent work has begun to evaluate *interaction patterns* within learning tasks. This work has revealed that fine-grained pattern analysis can shed light upon various cognitive, behavioral, and learning outcomes [21, 22, 29, 30, 37, 38]. For example, Lee and colleagues [21], and Liu and colleagues [22] evaluated how 3-step sequences of confusion [21, 22] and frustration [22] correlate to learning outcomes. Rodrigo and colleagues [29] also found that 3-step sequences of affective states (boredom, engaged concentration, confusion, and delight) from fine-grained detectors correlated to differences in learning outcomes. Sabourin and colleagues [30] found that the impact of student behavior on learning outcomes depended in part on the affect that preceded the behavior. Results from these studies reveal that fluctuations in students' affect and behavior over time (assessed through automated detectors) play important roles in learning outcomes.

However, much of this work had the limitation of only considering changes over brief periods of time. In this paper, we address this limitation by employing dynamical methodologies to quantify nuanced patterns of student affect, behavior, and learning across time, specifically two academic school years. We utilize fine-grained measures of affect, behavior, and knowledge (cognition) from middle school students who used the ASSISTments systems, and compute dynamical measures (i.e., Hurst and Entropy) of these constructs for each student. These measures (see below for details) characterize the occurrence and type of behavior across time for the constructs of interest (affect, behavior, knowledge) for each student within the ASSISTments environment.

We use two types of dynamical analysis techniques, Entropy and Hurst exponents. Entropy is a statistical measure used to assess the amount of predictability present within a time series [34]. Previously, Entropy has been used in EDM analyses by Snow and colleagues [38], to quantify the amount of randomness in students' interaction patterns within a game-based interface. Using this methodology they found that students who acted in more controlled (and predictable) manners had significantly higher task performance compared to students who acted in more random (or unpredictable) fashions. Hurst exponents are similar to Entropy in that they categorize the amount of order present within a system; however, unlike Entropy, Hurst exponents act as long-term correlations that capture how each moment in a time series

relates to the others. Thus, Hurst provides an even finer-grained look at the emergence of patterns across long periods of time. Recently, Hurst exponents have been used to characterize students' learning behaviors within game-based environments. For instance, Snow and colleagues [36] used this technique to examine nuanced fluctuations in students' choice patterns across time. Using the Hurst exponent, Snow and colleagues again found that students who acted in more deterministic manners (i.e., controlled and planned) were more likely to demonstrate higher learning gains compared to students who acted in more random (or impetuous) manners.

In the current work, we evaluate the degree to which Entropy and Hurst exponent measures based on affect, behavior, and knowledge (cognition) predicts a longer-term outcome, students' end-of-year state exam performance. This research was conducted on a dataset of 1,376 students who used ASSISTments when they were in middle school during the school years of 2004-2005 to 2005-2006 and took the standardized end-of-year state exams. We investigate in particular, the following research questions:

1) How are fluctuations in patterns of students' affect, behavior, and knowledge related to their end-of-year state math achievement test scores?

2) Are dynamical measures of affect, behavior, and knowledge predictive of student performance outcomes (end-of-year test score, i.e., MCAS)?

2. METHODOLOGY

2.1 Data Source: The ASSISTments System

This study explores students' learning outcomes and their interaction patterns from their usage of the ASSISTments system [27], a web-based tutoring system for middle-school mathematics, provided to students for free by Worcester Polytechnic Institute (WPI). As of 2013, ASSISTments has been used by over 50,000 students a year as part of their regular mathematics classes. ASSISTments *assesses* a student's knowledge while *assisting* them in learning, providing teachers with formative assessment of students as they progress in their acquisition of specific knowledge components.

Within the system, each problem maps to one or more cognitive skills. When students who are working on an ASSISTments problem answer correctly, they proceed to the next problem. When they answer incorrectly (Figure 1), the system scaffolds instruction by dividing the problem into component parts, stepping students through each before returning them to the original problem (as in Figure 2). Once the correct answer to the original question is provided, the student is prompted to go to the next question. Teachers use ASSISTments in designing problem sets completed by students either during class time or as homework assignments. ASSISTments provides data on student performance that is used by teachers to track misconceptions and discuss them in class.

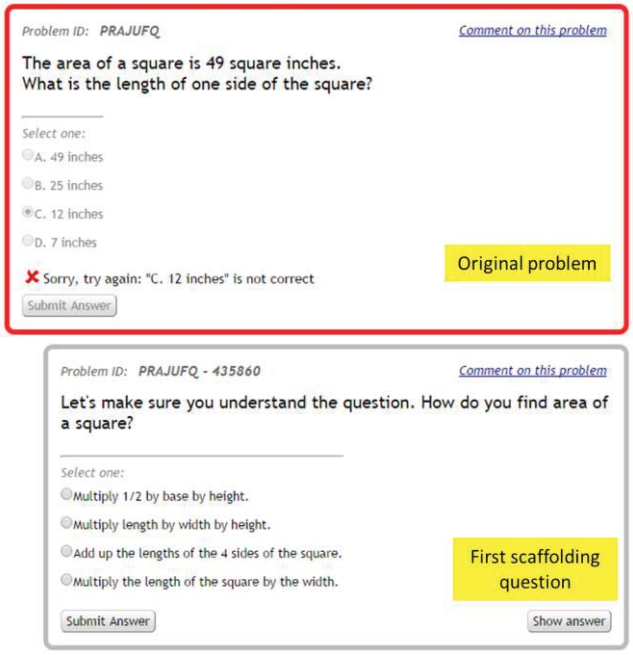


Figure 1. Example of an ASSISTments problem.

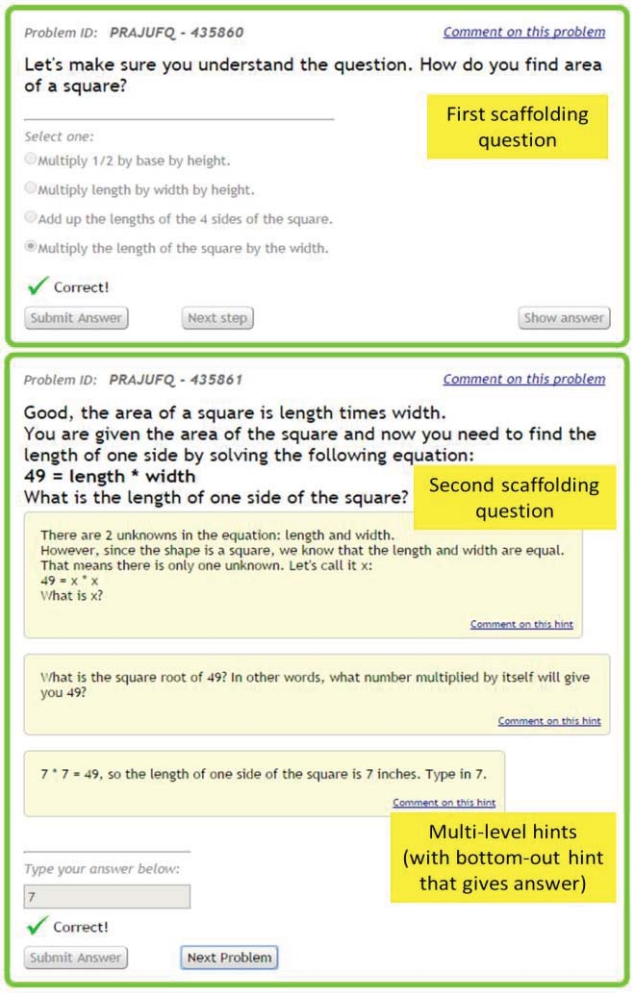


Figure 2. Example of Scaffolding and Hints in an ASSISTments Problem.

2.2 Data

2.2.1 State Exam Scores

Students who used ASSISTments when they were in middle school also took the MCAS (Massachusetts Comprehensive Assessment System) state standardized test near the end of their school years. The test is composed of English Language Arts, Mathematics and Science, and Technology subjects. This study analyzes usage of a tutoring system in mathematics; consequently, we examined the relationship of performance to the MCAS test scores for the math portion. Raw scores for the math portion range from 0 to 54 and are later scaled by the state after all tests have been scored. The scaled scores can be categorized into four groups: Failing, Needs Improvement, Proficient, and Advanced. Students in Massachusetts are required to score above failing to be able to graduate from high school; if students score in the Advanced group, they automatically earn a scholarship to a state college.

2.2.2 ASSISTments Data

Interaction log files from ASSISTments were obtained for 1,376 students who used the system when they were in middle school ranging from school years 2004-2005 to 2005-2006 (these school years were used due to the availability of the state exam data for these particular cohorts). These students, diverse in terms of both ethnicity and socio-economic status, were drawn from middle schools in an urban district in New England who used the ASSISTments system systematically during the school years. The 1,376 students generated a total of 830,167 actions within the system (an action may be answering a question, or requesting help), across around 3,700 original and scaffolding problems from ASSISTments, with an average of approximately 220 ASSISTments problems per student. Affect, behavior, and knowledge models were applied to this dataset to evaluate interaction patterns.

2.3 Computing Interaction Features

The interaction features used to compute dynamical assessments were generated using automated detectors of student engagement and learning previously developed and validated for ASSISTments. These included existing models of educationally-relevant affective states (boredom, engaged concentration, confusion, frustration), disengaged behaviors (off-task behavior and gaming the system), and student knowledge. Each of the detectors was applied to every action in the existing data set, in the same fashion as in previous publications [24]. We also included in our feature set of interactions, information on student correctness over time within ASSISTments.

2.3.1 Affect and Disengaged Behaviors

To obtain assessments of affect and disengaged behaviors, we leveraged existing detectors of student affect and behavior within the ASSISTments system [24]. Detectors of four affective states were utilized: boredom, engaged concentration, confusion, and frustration. Detectors of two disengaged behaviors are utilized: off-task behavior and gaming the system. Because our sample of students came from urban middle schools, their respective data were labeled using models optimized for students in urban schools [23, 24].

The affect and behavior detectors were developed in a two-stage process: first, student affect labels were acquired from field observations conducted using the BROMP protocol and HART Android app (reported in [24]), and then those labels were synchronized with the log files generated by ASSISTments at the same time. This process resulted in automated detectors that can be applied to log files at scale, specifically the data set used in this project (interaction log files for the 1,376 students). The detectors were constructed using only log data from student actions within the software occurring at the same time as or before the observations. The models performed as well as or better than other published models of sensor-free affect detection in educational software [3, 11, 13, 30]. They were then applied to the data set used in this paper to produce confidence values for each construct over time, which were then used to create dynamical assessments of affect and behavior.

2.3.2 Student Knowledge

Corbett and Anderson's [12] Bayesian Knowledge Tracing (BKT) model, a knowledge-estimation model that has been used in a considerable number of online learning systems, was applied to the data for this study. Models were fit by employing brute-force grid search (see [2]). BKT infers students' latent knowledge from their performance on problems that exercise the same set of skills. Each time a student attempts a problem or problem step for the first time, BKT recalculates the estimates of that student's knowledge for the skill (or knowledge component) involved in that problem. Estimations for each skill are made along four parameters: (1) L_O, the initial probability that the student knows the skill, (2) T, the probability of learning the skill at each opportunity to use that skill, (3) G, the probability that the student will give the correct answer despite not knowing the skill, and (4) S, the probability that the student will give an incorrect answer despite knowing the skill. The estimates obtained via BKT were calculated based on the student's first response to each problem, and were applied to each of the student's subsequent attempts on that problem.

We were able to distill interaction features –affect, behavior and knowledge using these models, as well as correctness – for each student action within the ASSISTments system. Affect and behavior features were initially computed at a 20-second grain-size and then applied to all relevant actions. These action-level features values are then used to compute student-level dynamical measures of Hurst and Entropy scores.

2.4 Dynamical Assessments of Student Interaction Features

Variations in students' interaction features (affect, behavior, knowledge, correctness) were assessed using two dynamical methodologies: Entropy analyses and Hurst exponents. These dynamic techniques are used to quantify (in standardized values) variations in students' interaction features and examine how these variations impacted students' year-end standardized test scores (i.e., MCAS). A description and explanation of Entropy analyses and Hurst exponents are described below.

2.4.1 Entropy

Entropy analyses were conducted to quantify the degree to which fluctuations in students' affective states were ordered (i.e., predictable) or disordered (i.e., unpredictable). Entropy analysis is a statistical measure that quantifies the overall tendency (i.e., amount of predictability) of a time series [34]. Entropy has been used across a variety of domains to measure random and ordered processes [15, 17, 34, 35, 38]. In the current study, Entropy is used to gain a deeper understanding of how changes in students' affective states across time may reflect ordered and disordered processes. To calculate Entropy, we applied the affect, behavior, and knowledge series produced from the models discussed above,

to data from school years 2004-2005 and 2005-2006. Entropy was then calculated using the following (standard) formula:

$$H(x) = -\sum_{i=0}^{N} P(x_i)(\log_e P(x_i)) \qquad (1)$$

Within the Entropy equation, $P(x_i)$ represents the probability of a given affective state. For instance, the Entropy for student X is the additive inverse of the sum of products calculated by multiplying the probability of each affect state by the natural log of the probability of that state. This formula affords the ability to capture the degree to which fluctuations in students' affect, behavior, knowledge, and correctness are ordered or disordered.

2.4.2 Hurst

While Entropy provides an overall quantification of a time series, it does not calculate how each moment in the time series may be related to the next. Thus, a more fine-grained analysis is needed to examine how fluctuations in students' affect, behavior, knowledge, and correctness manifest and change across time. To classify the tendency of students' affective states, Hurst exponents were calculated using Detrended Fluctuation Analysis (DFA) [26]. To calculate the Hurst exponent, the DFA integrates the normalized time series and then divides the series into equal intervals of length, *n*. Each interval is then fit with a least squares line and the integrated time series is *detrended* by subtracting the local predicted values (i.e., least square lines for each interval) from the integrated time series. The procedure is repeated for intervals of different lengths, increasing exponentially by the power of 2. Finally, each interval size is assigned a characteristic fluctuation, $F(n)$, that is calculated as the root mean square deviation of the integrated time series from local least squares lines. $\text{Log}_2 F(n)$ is then regressed onto $\log_2(n)$; which produces the slope of the regression line or Hurst exponent, *H*. Hurst exponents range from 0 to 1 and can be interpreted as follows: $0.5 < H \leq 1$ indicates persistent (controlled) behavior, $H = 0.5$ signifies random (independent) behavior, and $0 \leq H < 0.5$ denotes anti-persistent (reversion to the mean) behavior.

2.5 Predictive Modeling of State Test Scores

Prior work has shown that student usage choices while receiving tutoring in ASSISTments can predict as much of the variance in students' end-of-year state test scores as student performance can on items designed to assess test-related knowledge [16, 28]. It has also been shown that machine-learned and fine-grained assessments of affect and behavior can improve predictions of test score performance [24]. We extend this further and explore the value of also understanding the role of the degree of order/disorder of interaction (through occurrences of affect, behavior, knowledge, and correctness) in predicting student learning outcomes as reflected by students' end-of-year standardized examination scores.

After obtaining the aggregate student-level Hurst and Entropy scores for each student's patterns of affect, behavior, knowledge, and correctness, we examined how the degree of variation in the students' interaction patterns within ASSISTments was related to their MCAS math performance. We further examined these relations by conducting linear regression analyses on the students' MCAS math performance. We fit a cross-validated (6-fold, student-level) machine-learned model using linear regression with M5' feature selection to examine how students' dynamical assessments of interaction were predictive of their MCAS math scores. We generated reduced linear regression models that used three feature sets: (1) Hurst scores of interaction only, (2) Entropy scores of interaction only, and (3) both Hurst and Entropy scores of interaction. We then compared their cross-validated model performances and evaluated the features in the model with best performance values.

3. RESULTS

3.1 Hurst, Entropy, and State Test Scores

We first explore the relations between the MCAS scores for math and students' interaction patterns (i.e., their Hurst and Entropy scores) by examining the graphs of student proficiency (from MCAS performance) and the corresponding trends in Hurst and Entropy values. We grouped the students according to their scaled score groupings of Failing, Needs Improvement, Proficient, and Advanced, then computed for the average values of their Hurst and Entropy scores for affect, behavior, knowledge, and correctness in ASSISTments.

The graph of test proficiency and entropy measures (Figure 3) shows that low-achieving and high-achieving students experience fluctuations in affect, behavior, knowledge, and correctness while using ASSISTments in varying degrees. Students who have higher MCAS scores (i.e., *Advanced*) exhibited less fluctuation (lower entropy score) in their frustration ($F(3,1372) = 56.009$, $p < 0.001$, adjusted $\alpha = 0.013$), engaged concentration ($F(3,1372) = 27.334$, $p < 0.001$, adjusted $\alpha = 0.023$), off-task behavior ($\chi^2(3) = 64.089$, $p < 0.001$, adjusted $\alpha = 0.030$), and gaming the system ($\chi^2(3) = 238.350$, $p < 0.001$, adjusted $\alpha = 0.007$), but more fluctuation (higher entropy score) for boredom ($\chi^2(3) = 26.999$, $p < 0.001$, adjusted $\alpha = 0.040$), confusion ($\chi^2(3) = 29.759$, $p < 0.001$, adjusted $\alpha = 0.033$), correctness ($\chi^2(3) = 185.310$, $p < 0.001$, adjusted $\alpha = 0.010$), and knowledge ($\chi^2(3) = 639.111$, $p < 0.001$, adjusted $\alpha = 0.003$). [We used one-way ANOVA (F-test) for features with equal group variances, and Kruskal-Wallis test (χ^2 test) for features with unequal group variances.]

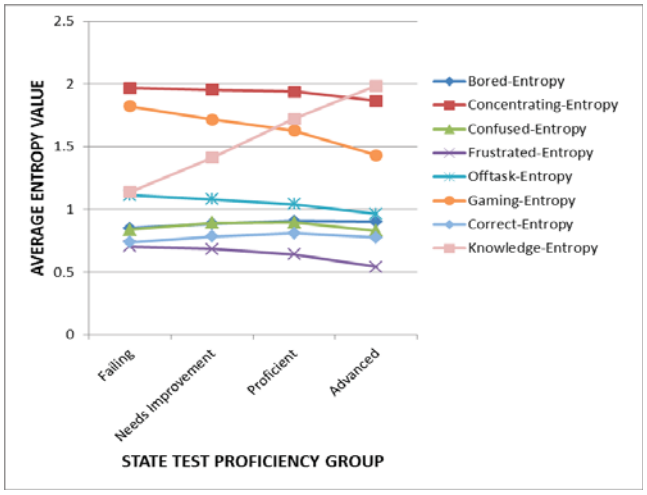


Figure 3. Entropy Scores by MCAS Test Score Category.

These trends suggest that students who performed better in MCAS showed overall consistency across time in exhibiting engaged concentration, frustration, off-task behaviors, and gaming the system, and an overall higher degree of variability across time in exhibiting boredom, confusion, correctness, and knowledge. It is possible that highly successful students may be more aware of their engaged concentration, frustration, off-task, and gaming behaviors within the system, compared to their awareness of the other constructs. Indeed, students who have achieved a higher level of proficiency or mastery of the material may also be more

efficient at controlling and maintaining the negative learning behaviors, and be more engaged. Interestingly, successful students show more variability, indicative of less control, in their boredom, confusion, correctness, and knowledge, possibly due to the nature of the learning task. These successful students may find some problems within ASSISTments too easy or too difficult with respect to their skills, causing them to experience varying degrees of boredom and confusion across time. In other words, the environment may be a major driver of the variability in these constructs. Another possibility comes from results in [24], where more successful students were more likely to be bored or confused when answering original problems, and less bored and confused when answering scaffolding problems. These successful students may also be overconfident in answering problems and become careless [32], exhibiting varying degrees of correctness and knowledge across time.

These relationships suggest that students with higher year-end exam scores were able to control their engagement by becoming less off-task and more consistent in overcoming their frustration and avoiding gaming the system, and be more engaged during their time in ASSISTments. However, a relevant area of future work may be to investigate whether the fluctuations across time for our interaction features are more a function of students' individual differences (e.g. proficiency) and their ability to control their learning behaviors [38], or a function of the learning task (e.g. type of problem, difficulty, etc.) and the learning behaviors it elicits from the students.

While Figure 3 shows the intensity or strength of fluctuations of our constructs across the entirety of student usage of ASSISTments, it does not demonstrate behavior of these fluctuations in fine-grained moments (i.e., persistence or anti-persistence of these constructs; how rapid were the fluctuations?). This is where looking at the Hurst measures of our constructs comes in useful. Figure 4 shows the graph of test proficiency and Hurst measures, where students who have higher MCAS scores achieved lower Hurst scores for engaged concentration ($\chi^2(3) = 134.719$, $p< 0.001$, adjusted $\alpha = 0.017$), frustration ($F(3,1372)= 27.543$, $p< 0.001$, adjusted $\alpha = 0.020$), off-task behavior ($\chi^2(3) = 70.736$, $p< 0.001$, adjusted $\alpha = 0.027$), and confusion ($F(3,1372)= 9.969$, $p< 0.001$, adjusted $\alpha = 0.037$), while higher Hurst scores for knowledge ($\chi^2(3) = 23.935$, $p< 0.001$, adjusted $\alpha = 0.043$) and gaming the system ($\chi^2(3) = 12.425$, $p= 0.006$, adjusted $\alpha = 0.047$).

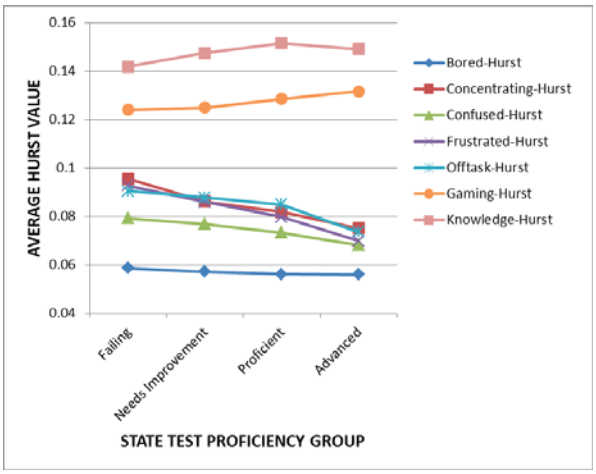


Figure 4. Hurst Scores by MCAS Test Score Category.

This trend in Hurst scores suggests that students who scored high on the MCAS had greater tendency to vary their behaviors, indicative of their actively adapting their learning behaviors. They instead showed regulation strategies in their ability to bounce back from frustration, resolve their confusion, and to re-engage after going off-task. Interestingly, more successful students show more mean reversion in engaged concentration than less successful students. Thus, more successful students were more variable in their engaged concentration (higher probability of concentration at one moment, lower probability of concentration on the next). Along with the Hurst scores for confusion, off-task and frustration, this Hurst trend for engaged concentration may indicate that students who began to feel confused or frustrated switched their focus and went off-task. Conversely, the trend for more successful students showed less variability in their display of knowledge and gaming the system behavior, which would suggest their ability to maintain their high level of knowledge and to not game the system. An understanding of the differences of rate of momentary fluctuations provides a lens on how students who vary in proficiency are able to effectively manage and adjust their affect, behavior, and knowledge within a learning task. It suggests that in the case of ASSISTments, it may be beneficial to teach less successful students strategies for quickly bouncing back from being off-task or ways to resolve their confusion and frustration.

We examine the significance of these differences in trends further by looking at the Pearson correlations between MCAS test scores and student Hurst and Entropy scores for affect, behavior, knowledge, and correctness (Table 1). We also utilize the Benjamini and Hochberg false discovery rate post-hoc correction to adjust the required alpha for significance and to reduce the occurrence of false positives, controlling for inflation of Type 1 error [8].

Table 1. Correlations with MCAS State Test Scores (- significant, p < 0.01; * - significant, p < 0.05)**

Hurst and Entropy Features	r	p-value	Adjusted α
Knowledge-Entropy	.705**	<0.001	0.003
Gaming-Entropy	-.441**	<0.001	0.007
Concentrating- Hurst	-.324**	<0.001	0.010
Frustration-Entropy	-.314**	<0.001	0.013
Correctness-Entropy	.275**	<0.001	0.017
Frustration- Hurst	-.252**	<0.001	0.020
Off-task-Entropy	-.211**	<0.001	0.023
Concentrating-Entropy	-.206**	<0.001	0.027
Off-task- Hurst	-.183**	<0.001	0.030
Confusion- Hurst	-.160**	<0.001	0.033
Bored-Entropy	.139**	<0.001	0.037
Bored-Hurst	-.100**	<0.001	0.040
Knowledge- Hurst	.076**	0.005	0.043
Confusion-Entropy	.076**	0.005	0.047
Gaming- Hurst	.059*	0.029	0.050
Correctness-Hurst	N/A	N/A	N/A

Table 1 shows that there are statistically significant, and reasonably strong relations between MCAS performance and Entropy measures of boredom, engaged concentration, confusion, frustration, off-task, gaming behavior, knowledge and correctness, and Hurst measures of boredom, engaged concentration, confusion, frustration, off-task, gaming behavior and knowledge. Note that for correctness only an Entropy score was calculated as it was a dichotomous measure of a student's answer (1 – correct, 0 – incorrect), and Hurst was not calculated for correctness as Hurst becomes less accurate when the inputs in the time series are discrete rather than continuous (which our other features are).

3.2 Prediction of State Test Scores

To examine the relations of these dynamical measures to MCAS performance, we conducted regression analyses to evaluate the predictive power of these measures (Table 2).

Table 2. State Test Score Model Performance Values Using Different Feature Sets (feature count is after feature selection)

Feature Set	R	R^2	RMSE	Number of Features
Hurst Features Only	0.400	0.160	11.251	5
Entropy Features Only	0.762	0.581	7.941	6
Both Hurst and Entropy Features	0.768	0.590	7.862	9

Combined, Hurst and Entropy assessments of affect, behavior, knowledge and correctness within ASSISTments are predictive of long-term performance (end-of-year state test score, MCAS) with reasonably high model performance. This finding shows that when our automated detectors of affect, behavior, and knowledge are applied at scale, the patterns generated are significantly related to learning outcomes. The specific patterns and contexts in which these interactions occur, however, remain to be further analyzed - for example using methods such as sequential pattern mining or recurrence analysis. Moreover, it is also worth noting that despite the interesting findings discussed above, the model created from dynamical assessments of machine-learned measures of interaction is not much better than a model created from just averaging our interaction features per student (for our sample, this model had a cross-validated R = 0.764) [24]. This suggests that averaging remains a good tool for predicting standardized exam scores, though it does not shed as much light on the phenomena of interest compared to the approach discussed here.

Optimized for predictor significance and model performance, our final model (Table 3) consists of either Hurst or Entropy scores (or both) of boredom, engaged concentration, confusion, frustration, gaming the system, knowledge, and correctness being predictive of MCAS performance.

Our final model leverages the relationships between MCAS and Hurst and entropy measures previously found. Stronger fluctuations across time for knowledge and correctness (positive coefficient for Entropy), and less persistence or quicker reversions in knowledge and engaged concentration (negative coefficient for Hurst), are associated with higher test scores for students. Furthermore, weaker fluctuations across time for boredom, confusion, gaming the system, and frustration (negative coefficient for Entropy), and more persistence or slow fluctuations for gaming the system (positive coefficient for Hurst), are associated with higher test scores for students. These relationships suggest that students with higher year-end exam scores were able to control their engagement by resolving their confusion, bouncing back from being bored, overcoming their frustration, and to show active learning, and be more consistent in not gaming the system during their time in ASSISTments.

Table 3. Final Model of Hurst and Entropy Scores Predicting State Test Scores

Predictors	B	Std. Error	t	Sig
(Constant)	28.821	3.258	8.845	<0.001
Correctness-Entropy	39.566	4.672	8.469	<0.001
Concentrating-Hurst	-34.185	10.738	-3.183	0.001
Gaming-Hurst	22.952	6.853	3.349	0.001
Knowledge-Hurst	-22.935	4.579	-5.009	<0.001
Bored-Entropy	-21.318	2.773	-7.687	<0.001
Frustration-Entropy	-17.874	1.892	-9.447	<0.001
Knowledge-Entropy	17.463	0.723	24.169	<0.001
Gaming-Entropy	-9.371	1.126	-8.320	<0.001
Confusion-Entropy	-6.157	1.803	-3.416	0.001

4. DISCUSSION AND CONCLUSION

In this paper, we utilized dynamical methodologies to investigate how nuanced patterns of affect, behavior, knowledge, and correctness were related to and predictive of students' end-of-year exam scores. Fine-grained models of student affect (boredom, engaged concentration, confusion, frustration) behavior (off-task behavior, gaming the system), and knowledge were applied to data from 1,376 students who used an educational software in mathematics over the course of a year during their middle school to generate interaction features. We then utilized dynamical measures of Hurst exponents and Entropy analysis to quantify the degree of randomness (or non-randomness) present within patterns of these interaction patterns.

Our results show that these dynamical assessments of students' interactions throughout the year (affect, behavior, knowledge, and correctness) are significantly associated with their end-of-year performance in a state test. Entropy scores of students for all of our interaction features showed significant differences between students in varied test proficiencies (as measured by the year-end exam). Across time, the more control a student demonstrated in frustration, engaged concentration, off-task behaviors, and gaming the system behaviors, as well as more flexibility in boredom, confusion, knowledge and correctness, the higher the student scored on the year-end exam. Students' Hurst scores also showed significant relations with the learning outcome, where students with more occurrences of fluctuations for engaged concentration, confusion, frustration, and off-task behaviors, and more persistence for knowledge and gaming the system were likely to perform better. These relations were supported by these dynamical assessments being predictive of performance in the end-of-year state test.

It is notable that most Hurst exponent values fell well below 0.5, indicating that overall, fine-grained machine-learned estimates of affect, behavior, knowledge in the system interaction of the 1,376 students are not random, and according to students' state or the learning task within the system, students show signs of switching between various degrees of affect, behavior, and knowledge over

time. In the future, it may be useful to examine sequential patterns of each interaction feature, looking also at the context and circumstances in the usage of the system that lead to students having increasing or decreasing occurrences (as well as points of inflection) in affect, behavior, and knowledge. The Hurst and Entropy may be able to be used in real-time to capture these affective changes and then provide feedback to a user model (or teacher) about the student. Less successful students may be made aware of their learning behaviors so they may more effectively regulate them, in particular for frustration, confusion, off-task-behavior, and gaming the system. They may also be taught strategies to more quickly bounce back from being off-task or even resolve their frustration and confusion.

Overall, these exploratory findings obtained when we dynamically assess the measures of interaction take a step further in evaluating how fine-grained machine-learned assessments of affect, behavior, and knowledge relate to learning outcomes. Looking at patterns using a combination of machine-learning techniques provides an avenue for observing the degree to which students regulate their actions in a learning task. Self-regulation research shows that when students are motivated to achieve learning goals they are more likely to regulate their behaviors [7]. This current study provides a preliminary lens on how dynamic measures of fine-grained series of distinctive affect (academic emotions) and behavior (engagement) are reflective of students' emotional and motivational regulation within a learning environment [9, 18], as well as the roles of affect and behavior on self-regulated learning [25].

5. ACKNOWLEDGMENTS

This research was supported by grants NSF #DRL-1031398, NSF #SBE-0836012, grant #OPP1048577 from the Bill and Melinda Gates Foundation, and grant #R305A130124 from the Institute of Education Sciences.

6. REFERENCES

[1] Baker, R.S.J.d. 2007. Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. In *Proceedings of ACM CHI 2007: Computer-Human Interaction*, 1059-1068.

[2] Baker R.S.J.d., Corbett A.T., Gowda S.M., Wagner A.Z., MacLaren B.M., Kauffman L.R., Mitchell A.P., and Giguere S. 2010. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP 2010*, 52-63.

[3] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G., Ocumpaugh, J., and Rossi, L. 2012. Towards Sensor-free Affect Detection in Cognitive Tutor Algebra. In *Proceedings of the 5th International Conference on Educational Data Mining*, 126-133.

[4] Baker, R.S., Corbett, A.T., and Koedinger, K.R. 2004. Detecting Student Misuse of Intelligent Tutoring Systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531-540.

[5] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, S.E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., and Beck, J. 2006. Adapting to When Students Game an Intelligent Tutoring System. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.

[6] Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., and Graesser, A. C. 2010. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4), 223-241.

[7] Bandura, A. 1991. Social cognitive theory of self-regulation. *Organizational behavior and human decision processes,* 50, 2, 248-287.

[8] Benjamini, Y. and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 289-300.

[9] Bosch, Nigel, and Sidney D'Mello. 2013. Sequential Patterns of Affective States of Novice Programmers. *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*.

[10] Cocea, M., Hershkovitz, A., and Baker, R.S.J.d. 2009. The Impact of Off-task and Gaming Behaviors on Learning: Immediate or Aggregate? In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, 507-514.

[11] Conati, C., and Maclaren, H. 2009. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction,* 19, 3, 267-303.

[12] Corbett, A.T., and Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction,* 4, 4, 253-278.

[13] D'Mello, S.K., Craig, S.D., Witherspoon, A. W., McDaniel, B. T., and Graesser, A. C. 2008. Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User- Adapted Interaction*, 18 (1-2), 45-80.

[14] D'Mello, S. K. and Graesser, A. C. 2012. Dynamics of Affective States during Complex Learning. *Learning and Instruction*, 22, 145-157.

[15] Fasolo, B., Hertwig, R., Huber, M., and Ludwig, M. 2009. Size, entropy, and density: What is the difference that makes the difference between small and large real-world assortments? *Psychology & Marketing*, 26, 3, 254-279.

[16] Feng, M., Heffernan, N.T., and Koedinger, K.R. 2009. Addressing the assessment challenge in an online system that tutors as it assesses. User Modeling and User-Adapted Interaction: *The Journal of Personalization Research (UMUAI)*, 19, 3, 243-266.

[17] Grossman, E. R. F. W. 1953. Entropy and choice time: The effect of frequency unbalance on choice-response. Quarterly *Journal of Experimental Psychology,* 41-51.

[18] Gumora, G. and Arsenio, W. F. 2002. Emotionality, emotion regulation, and school performance in middle school children. *Journal of School Psychology*, 40, 5, 395-413.

[19] Hadwin, A. F., Nesbit, J. C., Jamieson-Noel, D., Code, J., and Winne, P. H. 2007. Examining trace data to explore self-regulated learning. *Metaknowledge and Learning*, 2, 107–124.

[20] Hershkovitz, A., Baker, R.S.J.d., Gobert, J., Wixon, M., and Sao Pedro, M. 2013. Discovery with Models: A Case Study on Carelessness in Computer-based Science Inquiry. *American Behavioral Scientist*, 57, 10, 1479-1498.

[21] Lee, D. M. C., Rodrigo, M. M. T., d Baker, R. S., Sugay, J. O., and Coronel, A. 2011. Exploring the relationship between novice programmer confusion and achievement. In *Proceedings of Affective Computing and Intelligent Interaction*, 175-184.

[22] Liu, Z., Ocumpaugh, J., and Baker, R. S. 2013. Sequences of Frustration and Confusion, and Learning. In *Proc. Int. Conf. Ed. Data Mining*, 114-120.

[23] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., and Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*, 45, 3, 487-501.

[24] Pardos, Z.A., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2014. Affective states and state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *Journal of Learning Analytics*, 1, 107-128.

[25] Pekrun, R., Goetz, T., Titz, W., and Perry, R. P. 2002. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational Psychologist*, 37, 2, 91-105.

[26] Peng, C. K., Buldyrev, S. V., Havlin, S., Simons, H. E., Stanley, H. E., and Goldberger, A. L. 1994. Mosaic organization of DNA nucleotides. *Physical Review E*, 49, 1685-1689.

[27] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A. and Rasmussen, K.P. 2005. The Assistment project: Blending assessment and assisting. In *Proc. AIED 2005*, 555-562.

[28] Ritter, S., Joshi, A., Fancsali, S. E., and Nixon, T. 2013. Predicting Standardized Test Scores from Cognitive Tutor Interactions. In *Proceedings of the 6th International Conference on Educational Data Mining*, 169-176.

[29] Rodrigo, M. M. T., Baker, R. S., and Nabos, J. Q. 2010. The relationships between sequences of affective states and learner achievement. In *Proceedings of the 18th International Conference on Computers in Education*, 56-60.

[30] Sabourin, J., Mott, B., and Lester, J. 2011. Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks. In *Proc. ACII 2011*, 286-295.

[31] San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. 2013. Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In *Proceedings of the 6th International Conference on Educational Data Mining*, 177-184.

[32] San Pedro, M.O.Z., Baker, R.S.J.d., Rodrigo, Mercedes, M.M.T. 2014. Carelessness and Affect in an Intelligent Tutoring System for Mathematics. *International Journal of Artifiical Intelligence in Education*, 24, 189-210.

[33] San Pedro, M.O.Z., Ocumpaugh, J.L., Baker, R.S., Heffernan, N.T. 2014. Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In *Proceedings of the 7th International Conference on Educational Data Mining*, 276-279.

[34] Shannon, C. 1951. Prediction and Entropy of printed English. *Bell Systems Technical Journal*, 27, 50-64.

[35] Snow, E. L., Allen, L. K., Jacovina, M. E., and McNamara, D. S. 2015. Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers and Education,* 26, (2015), 378-392.

[36] Snow, E. L., Allen L. K., Russell, D. G., and McNamara, D. S. 2014. Who's in control?: Categorizing nuanced patterns of behaviors within a game-based intelligent tutoring system. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), Proce*edings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.

[37] Snow, E. L., Jackson, G. T., and McNamara, D. S. 2014. Emergent behaviors in computer-based learning environments: Computational signals of catching up. *Computers in Human Behavior,* 41, (2014), 62-70.

[38] Snow, E. L., Jacovina, M. E., Allen, L. K., Dai, J., and McNamara, D. S. 2014. Entropy: A stealth assessment of agency in learning environments. In J. Stamper, Z. Pardos, M. Mavrikis, B. M. McLaren (eds.), Proce*edings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.

Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection

Luc Paquette
Teachers College Columbia University
525 West 120th Street
New York, NY 10027
paquette@tc.columbia.edu

Jonathan Rowe
North Carolina State University
Raleigh, NC 27695
jprowe@ncsu.edu

Ryan Baker
Teachers College Columbia University
525 West 120th Street
New York, NY 10027
ryanshaunbaker@gmail.com

Bradford Mott
North Carolina State University
Raleigh, NC 27695
bwmott@ncsu.edu

James Lester
North Carolina State University
Raleigh, NC 27695
lester@ncsu.edu

Jeanine DeFalco
Teachers College Columbia University
525 West 120th Street
New York, NY 10027
jad2234@tc.columbia.edu

Keith Brawner
US Army Research Lab
Orlando, FL, USA
keith.w.brawner@mail.mil

Robert Sottilare
US Army Research Lab
Orlando, FL, USA
robert.sottilare@us.army.mil

Vasiliki Georgoulas
Teachers College Columbia University
525 West 120th Street
New York, NY 10027
vasiliki.georgoulas@usma.edu

ABSTRACT

Computational models that automatically detect learners' affective states are powerful tools for investigating the interplay of affect and learning. Over the past decade, affect detectors—which recognize learners' affective states at run-time using behavior logs and sensor data—have advanced substantially across a range of K-12 and postsecondary education settings. Machine learning-based affect detectors can be developed to utilize several types of data, including software logs, video/audio recordings, tutorial dialogues, and physical sensors. However, there has been limited research on how different data modalities combine and complement one another, particularly across different contexts, domains, and populations. In this paper, we describe work using the Generalized Intelligent Framework for Tutoring (GIFT) to build multi-channel affect detection models for a serious game on tactical combat casualty care. We compare the creation and predictive performance of models developed for two different data modalities: 1) software logs of learner interactions with the serious game, and 2) posture data from a Microsoft Kinect sensor. We find that interaction-based detectors outperform posture-based detectors for our population, but show high variability in predictive performance across different affect. Notably, our posture-based detectors largely utilize predictor features drawn from the research literature, but do not replicate prior findings that these features lead to accurate detectors of learner affect.

Keywords

Affect detection, multimodal interaction, posture, serious games.

1. INTRODUCTION

Affect is critical to understanding learning. However, the interplay between affect and learning is complex. Some affective states, such as boredom, have been shown to coincide with reduced learning outcomes ([25]). Other affective states, such as confusion and engaged concentration, have been found to serve beneficial roles ([14], [24]). The ability to detect a learner's affective state while she interacts with an online learning environment is critical for adaptive learning technologies that aim to support and regulate learners' affect ([26]).

Research on affective computing has enabled the development of models that automatically detect learner affect using a wide variety of data modalities (see extensive review in [8]). Many researchers have focused on physical sensors, because of their capacity to capture physiological and behavioral manifestations of emotion, potentially regardless of what learning system is being used. Sensor-based detectors of affect have been developed using a range of physical indicators including facial expressions ([2], [7]), voice [35], posture ([11], [16]), physiological data [22] and EEG [1]. Despite this promise, deploying physical sensors in the classroom is challenging, and sometimes prohibitive [6], and efforts in this area are still ongoing, with some researchers arguing that this type of affect detection has not yet reached its full potential [13].

In recent years, efforts have also been made towards the development of complementary affect detection techniques that recognize affect solely from logs of learner interactions with an online learning environment ([2], [3], [24]). Initial results in this area have shown considerable promise. As both sensor-based and interaction-based affect detectors continue to mature, efforts are needed to compare the relative advantages of each approach. An early comparison was seen in D'Mello et al. [15], but considerable progress has been made in the years since.

In this paper, we compare the performance and the general process of developing models for affect detection using two different data modalities: learner interaction logs and posture data

from a Microsoft Kinect sensor. Ground-truth affect data for detector development was collected through field observation [23] of learners interacting with vMedic, a serious game on tactical combat casualty care, integrated into the General Intelligent Framework for Tutoring (GIFT) [32]. Findings suggest that interaction-based affect detectors outperform posture-based detectors for our population. However, interaction-based detectors show high variability in predictive performance across different emotions. Further, our posture-based detectors, which utilize many of the same predictor features found throughout the research literature, achieve predictive performance that is only slightly better than chance across a range of affective states, a finding that is contrary to prior work on sensor-based affect detection.

2. DATA

Three sources of data were used in this work: 1) log file data produced by learners using the vMedic (a.k.a. TC3Sim) serious game, 2) Kinect sensor log data, and 3) quantitative field observations of learner affect using the BROMP 1.0 protocol [23]. This section describes those sources of data, by providing information on the learning environment, study participants, and research study method.

2.1 Learning System and Subjects

We modeled learner affect within the context of vMedic, a serious game used to train US Army combat medics and lifesavers on tasks associated with dispensing tactical field care and care under fire (Figure 1). vMedic has been integrated with the Generalized Intelligent Framework for Tutoring (GIFT) [32], a software framework that includes a suite of tools, methods, and standards for research and development on intelligent tutoring systems and affective computing.

Game-based learning environments, such as vMedic, enable learners to interact with virtual worlds, often through an avatar, and place fewer constraints on learner actions than many other types of computer-based learning environments ([3], [19], [24]). Some virtual environments place more constraints on learner behavior than others. For example, learning scenarios in vMedic are structured linearly, presenting a fixed series of events regardless of the learner's actions. In contrast, game-based learning environments such as EcoMUVE [20] and Crystal Island [29] afford learners considerable freedom to explore the virtual world as they please. While vMedic supports a considerable amount of learner control, its training scenarios focus participants' attention on the objectives of the game (e.g., administering care), implicitly guiding learner experiences toward key learning objectives.

To investigate interaction-based and sensor-based affect detectors for vMedic, we utilize data from a study conducted at the United States Military Academy (USMA). There were 119 cadets who participated in the study (83% male, 17% female). The participants were predominantly first-year students. During the data collection, all participants completed the same training module. The training module focused on a subset of skills for tactical combat casualty care: care under fire, hemorrhage control, and tactical field care. The study materials, including pre-tests, training materials, and post-tests, were administered through GIFT. At the onset of each study session, learners completed a content pre-test on tactical combat casualty care. Afterward, participants were presented with a PowerPoint presentation about tactical combat casualty care. After completing the PowerPoint, participants completed a series of training scenarios in the vMedic serious game where they applied skills, procedures, and knowledge presented in the PowerPoint. In vMedic, the learner adopts the role of a combat medic faced with a situation where one (or several) of her fellow soldiers has been seriously injured. The learner is responsible for properly treating and evacuating the casualty, while following appropriate battlefield doctrine. After the vMedic training scenarios, participants completed a post-test, which included the same series of content assessment items as the pre-test. In addition, participants completed two questionnaires about their experiences in vMedic: the Intrinsic Motivation Inventory (IMI) [30] and Presence Questionnaire [34]. All combined study activities lasted approximately one hour.

During the study, ten separate research stations were configured to collect data simultaneously; each station was used by one cadet at a time. Each station consisted of an Alienware laptop, a Microsoft Kinect for Windows sensor, and an Affectiva Q-Sensor, as well as a mouse and pair of headphones. The study room's layout is shown in Figure 2. In the figure, participant stations are denoted as ovals. Red cones show the locations of Microsoft Kinect sensors, as well as the sensors' approximate fields of view. The dashed line denotes the walking path for the field observers.

Kinect sensors recorded participants' physical behavior during the study, including head movements and posture shifts. Each Kinect sensor was mounted on a tripod and positioned in front of a participant (Figure 2). The Kinect integration with GIFT provided four data channels: skeleton tracking, face tracking, RGB (i.e., color), and depth data. The first two channels leveraged built-in tracking algorithms (which are included with the Microsoft Kinect for Windows SDK) for recognizing a user's skeleton and face, each represented as a collection of 3D vertex coordinates. The RGB channel is a 640x480 color image stream comparable to a standard web camera. The depth channel is a 640x480 IR-based image stream depicting distances between objects and the sensor.

Q-Sensors recorded participants' physiological responses to events during the study. The Q-Sensor is a wearable arm bracelet that measures participants' electrodermal activity (i.e., skin conductance), skin temperature, and its orientation through a built-in 3-axis accelerometer. However, Q-Sensor logs terminated prematurely for a large number of participants, necessitating additional work to determine the subset of field observations that are appropriate to predict with Q-Sensor-based features. Inducing Q-Sensor-based affect detectors will be an area of future work.

Figure 1. vMedic learning environment.

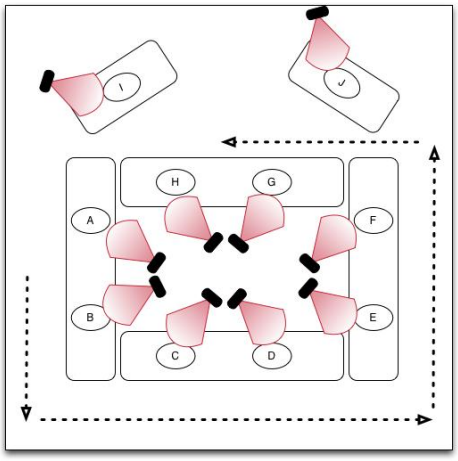


Figure 2. Study room layout.

2.2 Quantitative Field Observations (QFOs)

We obtain ground-truth labels of affect using Quantitative Field Observations (QFOs), collected using the Baker-Rodrigo-Ocumpaugh Monitoring Protocol (BROMP) [23]. This is a common practice for interaction-based detection of affect (e.g. [3], [24]). Much of the work to date for video-based affect detection, by contrast, has focused on modeling emotion labels that are based on self-reports ([10], [16]), or labels obtained through retrospective judgments involving freeze-frame video analysis [11]. It has been argued that BROMP data is easier to obtain and maintain reliability for under real-world conditions than these alternate methods [23], being less disruptive than self-report, and easier to gain full context than video data.

To be considered BROMP-certified, a coder must achieve inter-rater reliability of Kappa >= 0.6 with a previously BROMP-certified coder. BROMP has been used for several years to study behavior and affect in educational settings ([3], [4], [27]), with around 150 BROMP-certified coders as of this writing, and has been used as the basis for successful automated detectors of affect ([3], [24]). Observations in this study were conducted by two BROMP-certified coders, the 2nd and 6th authors of this paper.

Within the BROMP protocol, behavior and affective states are coded separately but simultaneously using the Human Affect Recording Tool (HART), an application developed for the Android platform (and freely available as part of the GIFT distribution). HART enforces a strict coding order determined at the beginning of each session. Learners are coded individually, and coders are trained to rely on peripheral vision and side glances in order to minimize observer effects. The coder has up to 20 seconds to categorize each trainee's behavior and affect, but records only the first thing he or she sees. In situations where the trainee has left the room, the system has crashed, where his or her affect or behavior do not match any of the categories in the current coding scheme, or when the trainee can otherwise not be adequately observed, a '?' is recorded, and that observation is eliminated from the training data used to construct automated detectors.

In this study, the typical coding scheme used by BROMP was modified to accommodate the unique behaviors and affect that was manifest for this specific cadet population and domain. Affective states observed included frustration, confusion, engaged concentration, boredom, surprise and anxiety. Behavioral categories consisted of on-task, off-task behaviors, Without Thinking Fastidiously behavior [33], and intentional friendly fire (these last two categories will not be discussed in detail, as they were rare).

In total, 3066 BROMP observations were collected by the two coders. Those observations were collected over the full length of the cadets' participation in the study, including when they were answering questionnaires on self-efficacy, completing the pre and post-tests, reviewing PowerPoint presentations, and using vMedic. For this study, we used only the 755 observations that were collected while cadets were using vMedic. Of those 755 observations, 735 (97.35%) were coded as the cadet being on-task, 19 (2.52%) as off-task, 1 (0.13%) as Without Thinking Fastidiously, and 0 as intentional friendly fire. Similarly, 435 (57.62%) of the affect labels were coded as concentrating, 174 (23.05%) as confused, 73 (9.67%) as bored, 32 (4.24%) as frustrated, 29 (3.84%) as surprised and 12 (1.59%) as anxious.

3. INTERACTION-BASED DETECTORS

The BROMP observations collected while cadets were using vMedic were used to develop machine-learned models to automatically detect the cadet's affective states. In this section, we discuss our work to develop affect detectors based on cadets' vMedic interactions logs.

3.1 Data Integration

In order to generate training data for our interaction-based affect detectors, trainee actions within the software were synchronized to field observations collected using the HART application. During data collections, both the handheld computers and the GIFT server were synchronized to the same internet NTP time server. Timestamps from both the HART observations and the interaction data were used to associate each observation to the actions that occurred during the 20 seconds window prior to data entry by the observer. Those actions were considered as co-occurring with the observation.

3.2 Feature Distillation

For each observation, we distilled a set of 38 features that summarized the actions that co-occurred with or preceded that observation. Those features included: changes in the casualty, both recent and since injury, such as changes in blood volume, bleed rate and heart rate; player states in terms of attacker, such as being under cover and being with the unit; the number of time specific actions, such as applying a tourniquet or requesting a security sweep, were executed; and time between actions. (see [5] for a more complete list of features.)

3.3 Machine Learning Process

Detectors were built separately for each affective state and behavioral constructs. For example a detector was used to distinguish observations of boredom from observations that were not boredom. It is worth noting that the construct of engaged concentration, was defined during modeling as a learner having the affect of concentration and not being off-task, since concentrating while being off-task reflects concentration with something other than learning within the vMedic game. Only 2 such observations was found amongst the collected observations. Detectors were not developed for off-task behavior, Without Thinking Fastidiously behavior, and anxiety due to the low number of observations for those construct (19, 1 and 12 respectively).

Each detector was validated using 10-fold participant-level cross-validation. In this process, the trainees are randomly separated into 10 groups of equal size and a detector is built using data for

each combination of 9 of the 10 groups before being tested on the 10th group. By cross-validating at this level, we increase confidence that detectors will be accurate for new trainees. Oversampling (through cloning of minority class observations) was used to make the class frequency more balanced during detector development. However, performance calculations were made with reference to the original dataset.

Detectors were fit in RapidMiner 5.3 [21] using six machine learning algorithms that have been successful for building similar detectors in the past ([3], [24]): J48, JRip, NaiveBayes, Step Regression, Logistic Regression and KStar. The detector with the best performance was selected for each affective state. Detector performance was evaluated using two metrics: Cohen's Kappa [9] and A' computed as the Wilcoxon statistic [18]. Cohen's Kappa assesses the degree to which the detector is better than chance at identifying the modeled construct. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly. A' is the probability that the algorithm will correctly identify whether an observation is a positive or a negative example of the construct (e.g. is the learner bored or not?). A' is equivalent to the area under the ROC curve in signal detection theory [18]. A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly. A' was computed at the observation level.

When fitting models, feature selection was performed using forward selection on the Kappa metric. Performance was evaluated by repeating the feature selection process on each fold of the trainee-level cross-validation in order to evaluate how well models created using this feature selection procedure perform on new and unseen test data. The final models were obtained by applying the feature selection to the complete dataset.

4. POSTURE-BASED DETECTORS

The second set of affect detectors we built were based on learner posture during interactions with vMedic. Kinect sensors produced data streams that were utilized to determine learner posture. Using machine learning algorithms, we trained models to recognize affective states based on postural features.

4.1 Data Integration

GIFT has a *sensor module* that is responsible for managing all connected sensors and associated data streams. This includes Kinect sensor data, which is comprised of four complementary data streams: face tracking, skeleton tracking, RGB channel, and depth channel data. Face- and skeleton-tracking data are written to disk in CSV format, with rows denoting time-stamped observations and columns denoting vertex coordinates. RGB and depth channel data are written to disk as compressed binary data files. To analyze data from the RGB and depth channels, one must utilize the GiftKinectDecoder, a standalone utility that is packaged with GIFT, to decompress and render the image data into a series of images with timestamp-based file names. Data from all four channels can be accessed and analyzed outside of GIFT. For the present study, we utilized only vertex data to analyze participants' posture. Each observation in the vertex data consisted of a timestamp and a set of 3D coordinates for 91 vertices, each tracking a key point on the learner's face (aka face tracking) or upper body (aka skeletal tracking). The Kinect sensor sampled learners' body position at a frequency of 10-12 Hz.

It was necessary to clean the Kinect sensor data in order to remove anomalies from the face and skeletal tracking. Close examination of the Kinect data revealed periodic, and sudden, jumps in the coordinates of posture-related vertices across frames. These jumps were much larger than typically observed across successive frames, and they occurred due to an issue with the way GIFT logged tracked skeletons: recording the *most recently* detected skeleton, rather than the *nearest* detected skeleton. This approach to logging skeleton data caused GIFT to occasionally log bystanders standing in the Kinect's field of view rather than the learner using vMedic. In our study, such a situation could occur when a field observer walked behind the trainee.

To identify observations that corresponded to field observers rather than participants, Euclidean distances between subsequent observations of a central vertex were calculated. The distribution of Euclidean distances was plotted to inspect the distribution of between-frame movements of the vertex. If the Kinect tracked field observers, who were physically located several feet behind participants, the distribution was likely to be bimodal. In this case, one cluster would correspond to regular posture shifts of a participant between frames, and the other cluster corresponded to shifts between tracking participants and field observers. This distribution could be used to identify a distance threshold for determining which observations should be thrown out, as they were likely due to tracking field observers rather than participants. Although the filtering process was successful, the need for this process reveals a challenge to the use of BROMP for detectors eventually developed using Kinect or video data.

In addition to cleaning the face and skeleton mesh data, we performed a filtering process to remove data that were unnecessary for the creation of posture-based affect detectors. A majority of the facial vertices recorded by the Kinect sensor were not necessary for investigating trainees' posture. Of the 91 vertices recorded by the Kinect sensor, only three were utilized for posture analysis: *top_skull*, *head*, and *center_shoulder*. These vertices were selected based on prior work investigating postural indicators of emotion with Kinect data [16].

Finally, HART observations were synchronized with the data collected from the Kinect sensor. As was the case for our interaction-based sensor, the Kinect data provided by GIFT was synchronized to the same NTP time server as the HART data. This allowed us to associate field observations with observations of face and skeleton data produced by the Kinect sensor.

4.2 Feature Distillation

We used the Kinect face and skeleton vertex data to compute a set of predictor features for each field observation. The engineered features were inspired by related work on posture sensors in the affective computing literature, including work with pressure-sensitive chairs ([10], [11]) and, more recently, Kinect sensors [16]. Several research groups have converged on common sets of postural indicators of emotional states. For example, in several cases boredom has been found to be associated with leaning back, as well as increases in posture variance ([10], [11]). Conversely, confusion and flow have been found to be associated with forward-leaning behavior ([10], [11]).

We computed a set of 73 posture-related features. The feature set was designed to emulate the posture-related features that had previously been utilized in the aforementioned posture-based affect detection work ([10], [11], [16], [17]). For each of three retained skeletal vertices tracked by the Kinect (*head*, *center_shoulder*, and *top_skull*), we calculated 18 features based on multiple time window durations. These features are analogous to those described in [16], and were previously found to predict learners' retrospective self-reports of frustration and engagement:

- Most recently observed distance

Table 1. Performance of each of the interaction-based and posture-based detectors of affect

Affect	Interaction-Based Detectors			Posture-Based Detectors		
	Classifier	**Kappa**	**A'**	**Classifier**	**Kappa**	**A'**
Boredom	Logistic Regression	0.469	0.848	Logistic Regression	0.109	0.528
Confusion	Naïve Bayes	0.056	0.552	JRip	0.062	0.535
Engaged Concentration	Step Regression	0.156	0.590	J48	0.087	0.532
Frustration	Logistic Regression	0.105	0.692	Support Vec. Machine	0.061	0.518
Surprise	KStar	0.081	0.698	Logistic Regression	-0.001	0.493

- Most recently observed depth (Z coordinate)
- Minimum observed distance observed thus far
- Maximum observed distance observed thus far
- Median observed distance observed thus far
- Variance in distance observed thus far
- Minimum observed distance during past 5 seconds
- Maximum observed distance during past 5 seconds
- Median observed distance during past 5 seconds
- Variance in distance during past 5 seconds
- Minimum observed distance during past 10 seconds
- Maximum observed distance during past 10 seconds
- Median observed distance during past 10 seconds
- Variance in distance during past 10 seconds
- Minimum observed distance during past 20 seconds
- Maximum observed distance during past 20 seconds
- Median observed distance during past 20 seconds
- Variance in distance during past 20 seconds

We also induced several *net_change* features, which are analogous to those reported in [11] and [10] using pressure-sensitive seat data:

$$net\_dist\_change[t] = \left| \begin{array}{c} head\_dist[t] - head\_dist[t-1] + \\ cen\_shldr\_dist[t] - cen\_shldr\_dist[t-1] + \\ top\_skull\_dist[t] - top\_skull\_dist[t-1] \end{array} \right| \quad (1)$$

$$net\_pos\_change[t] = \left| \begin{array}{c} head\_pos[t] - head\_pos[t-1] + \\ cen\_shldr\_pos[t] - cen\_shldr\_pos[t-1] + \\ top\_skull\_pos[t] - top\_skull\_pos[t-1] \end{array} \right| \quad (2)$$

These features were calculated from Kinect vertex tracking data, as opposed to seat pressure data. Specifically, the *net_dist_change* feature was calculated as each vertex's net change in distance (from the Kinect sensor) over a given time window, and then summed together. The *net_pos_change* feature was calculated as the Euclidean distance between each vertex's change in position over a given time window, and then summed together. Both the *net_dist_change* feature and *net_pos_change* feature were calculated for 3 second and 20 second time windows.

We also calculated several *sit_forward*, *sit_back*, and *sit_mid* features analogous to [10] and [17]. To compute these features, we first calculated the average median distance of participants' *head* vertex from each Kinect sensor. This provided a median distance for each of the 10 study stations (see Figure 1). We also calculated the average standard deviation of *head* distance from each sensor. Then, based on the station-specific medians and standard deviations, we calculated the following features for each participant:

$$sit\_forward = \begin{cases} 1 & if\ head\_dist \le median\_dist - st\_dev \\ 0 & otherwise \end{cases} \quad (3)$$

$$sit\_back = \begin{cases} 1 & if\ head\_dist \ge median\_dist + st\_dev \\ 0 & otherwise \end{cases} \quad (4)$$

The *sit_mid* feature was the logical complement of *sit_forward* and *sit_back*; if a learner was neither sitting forward, nor sitting back, they were considered to be in the *sit_mid* state. We also computed predictor features that characterized the proportion of observations in which the learner was in a *sit_forward*, *sit_back*, or *sit_mid* state over a window of time. Specifically, we calculated these features for 5, 10, and 20 second time windows, as well as over the entire session to-date.

4.3 Machine Learning

Posture-based detectors of affect were built using a process analogous to the one used to build our interaction-based detectors. As such, separate detectors were, once again, built for each individual affective state and behavioral construct. All observations labeled as '?' were removed from the training set as they represent observations where the cadet's affective state or behavior could not be determined.

Each detector was validated using 10-fold participant-level cross-validation. Oversampling was used to balance class frequency by cloning minority class instances, as was the case when training our interaction-based detectors. RapidMiner 5.3 was used to train the detectors using multiple different classification algorithms: J48 decision trees, naïve Bayes, support vector machines, logistic regression, and JRip. When fitting posture-based affect detection models, feature selection was, once again, performed through forward selection using a process analogous to the one used for our interaction-based detectors.

5. RESULTS

As discussed above, each of the interaction-based and posture-based detectors of affect were cross-validated at the participant level (10 folds) and performance was evaluated using both Kappa and A'. Table 1 summarizes the performance achieved by each detector for both the Kappa and A' metrics.

Performance of our interaction-based detectors was highly variable across affective states. The detector of boredom achieved, by far, the highest performance (Kappa = 0.469, A' = 0.848) while some of the other detectors achieved very low performance. This was the case for the confusion detector that performed barely above chance level (Kappa = 0.056, A' = 0.552). Detectors of

frustration and surprise achieved relatively low Kappa (0.105 and 0.081 respectively), but good A' (0.692 and 0.698 respectively). Performance for engaged concentration achieved a Kappa closer to the average (0.156), but below average A' (0.590).

In general, posture-based detectors performed only slightly better than chance, with the exception of the surprise detector, which actually performed worse than chance. The boredom detector, induced as a logistic regression model, achieved the highest predictive performance (Kappa = 0.109, A' = 0.528), induced as a logistic regression model.

6. DISCUSSION

Across affective states, the posture-based detectors achieved lower predictive performance than the interaction-based detectors. In fact, the posture-based detectors performed only slightly better than chance, and in the case of some algorithms and emotions, worse than chance. This finding is notable, given that our distilled posture features were inspired largely from the research literature, where these types of features have been shown to predict learner emotions effectively in other contexts ([10], [11], [16], [17]). For example, D'Mello and Graesser found machine-learned classifiers discriminating affective states from neutral yielded kappa values of 0.17, on average [10]. Their work utilized posture features distilled from pressure seat data, including several features analogous to those used in our work. Grafsgaard et al. found that Pearson correlation analyses with retrospective self-reports of affect revealed significant relationships between posture and emotion, including frustration, focused attention, involvement, and overall engagement. Reported correlation coefficients ranged in magnitude from 0.35 to 0.56, which are generally considered moderate to large effects [19]. Cooper et al. found that posture seat-based features were particularly effective for predicting excitement in stepwise regression analyses ($R = 0.56$), and provided predictive benefits beyond log-based models across a range of emotions [10]. While the methods employed in each of these studies differ from our own, and thus the empirical results are not directly comparable, the qualitative difference in the predictive value of postural features is notable.

There are several possible explanations for why our posture-based predictors were not more effective. First, our use of BROMP to generate affect labels distinguishes our work from prior efforts, which used self-reports ([10], [16], [17]) or retrospective video freeze-frame analyses [11]. It is possible that BROMP-based labels of affect present distinct challenges for posture-based affect detection. BROMP labels are based on holistic judgments of affect, and pertain to 20-second intervals of time, which may be ill matched for methods that depend upon low-level postural features to predict emotion. Similarly, much of the work on posture-based affect detection has taken place in laboratory settings involving a single participant at a time [11], especially prior work using Kinect sensors ([16], [17]). In contrast, our study was performed with up to 10 simultaneous participants (see Figure 2), introducing potential variations in sensor positions and orientations. This variation may have introduced noise to our posture data, making the task of inducing population-general affect detectors more challenging than in settings where data is collected from a single sensor. If correct, this explanation underscores the challenges inherent in scaling and generalizing sensor-based affect detectors.

The study room's setup also limited how sensors could be positioned and oriented relative to participants. For example, it was not possible to orient Kinect cameras to the sides of participants, capturing participants' profiles, which would have made it easier to detect forward-leaning and backward-leaning postures. This approach has shown promise in other work, but was not a viable option in our study [31]. Had the Kinect sensors been positioned in this manner, the video streams would have been disrupted by other participants' presence in the cameras' fields of view.

Another possible explanation has to do with the population of learners that was involved in the study: U.S. Military Academy (USMA) cadets. Both BROMP observers noted that the population's affective expressiveness was generally different in kind and magnitude than the K-12 and civilian academic populations they were more accustomed to studying. Specifically, they indicated that the USMA population's facial and behavioral expressions of affect were relatively subdued, perhaps due to military cultural norms. As such, displays of affect via movement and body language may have been more difficult to recognize than would have otherwise been encountered in other populations.

In general, we consider the study population, BROMP affect labels, and naturalistic research setup to be strengths of the study. Indeed, despite the difference in how military display affect compared to the K-12 and civilian academic population, human observers were able to achieve the inter-rater reliability required by BROMP (Kappa >= 0.6) [23]. Thus we do not have plans to change these components in future work. Instead, we will likely seek to revise and enhance the data mining techniques that we employ to recognize learner affect, as well as the predictor features engineered from raw posture data. In addition, we plan to explore the predictive utility of untapped data streams (e.g., Q-Sensor data, video data).

It is notable that our interaction-based detectors had a more varied performance than had been seen in prior studies using this methodology; the detectors were excellent for boredom, and varied from good to just above chance for other constructs. It is possible that this too is due to the population studied, but may also be due to the nature of the features that were distilled in order to build the models. For example, the high performance of our detector of boredom can be attributed to the fact that one feature, whether the student executed any meaningful actions in the 20 second observation window, very closely matched the trainees' manifestation of this affective state. In fact, a logistic regression detector trained using this feature alone achieved higher performance than our detectors for any of the other affective state (Kappa = 0.362, A' = 0.680). It can be difficult to predict, a priori, which features will most contribute to the detection of a specific affective state. It is also possible that some of the affective states for which interaction-based detection was less effective (e.g., confusion) simply did not manifest consistently in the interactions with the learning environment across different trainees. It is thus difficult to determine whether poor performance of detectors for some constructs, such as our confusion detector, is due to insufficient feature engineering or inconsistent behaviors by the trainee. As such, the creation of interaction-based detectors is an iterative process, where features are engineered, and models are induced and refined, until performance reaches an acceptable level, or no improvement in performance is observed, despite repeated knowledge-engineering efforts.

We aim to identify methods to improve the predictive accuracy of posture-based detectors in future work. One advantage they possess relative to interaction-based detectors is that posture-based detectors may be more generalizable, since they pertain to aspects of learner behavior that are outside of the software itself. By contrast, much of the effort invested in the creation of interaction-based detectors is specific to the system for which the

detectors are created. Features are built to summarize the learner's interaction in the learning environment and, as such, are dependent on the system's user interface. Much of the creation of interaction-based detectors must hence be replicated for new learning environments, though there have been some attempts to build toolkits that can replicate features seen across many environments, such as unitizing the time between actions by the type of action or problem step (e.g. [28]).

On the other hand, posture-based detectors are built upon a set of features that are more independent of the system for which the detectors are designed. The process of creating the features itself requires considerable effort when compared with building a set of features for interaction-based detectors, such as elaborate efforts to adequately clean the data, but at least in principle, it is only necessary to develop the methods for doing so once. The same data cleaning and feature distillation procedures can be repeated for subsequent systems. This is especially useful in the context of a generalized, multi-system tutoring framework such as GIFT [32]. Although different posture-based affect detectors might need to be created for different tutoring systems—due to differences in the postures associated with affect for different populations of learners, environments and contexts—the posture features we computed from the data provided by Kinect sensors will ultimately become available for re-use by any tutor created using GIFT. This has the potential to considerably reduce the time required to build future posture-based affect detectors for learning environments integrated with the GIFT architecture.

7. CONCLUSION

Interaction-based and posture-based detectors of affect show considerable promise for adaptive computer-based learning environments. We have investigated their creation and predictive performance in the context of military cadets using the vMedic serious game for tactical combat casualty care. Interaction-based and posture-based detectors capture distinct aspects of learners' affect. Whereas interaction-based detectors capture the relationship between affect and its impact on the trainee's action in the learning environment, posture-based detectors capture learners' physical expressions of emotion.

In our study, we found that interaction-based detectors achieved overall higher performance than posture-based detectors. We speculate that the relatively weak predictive performance of our posture-based affect detectors may be due to some combination of the following: the interplay of high-level BROMP affect labels and low-level postural features, the challenges inherent in running sensor-based affect studies with multiple simultaneous participants, and population-specific idiosyncrasies in USMA cadets' affective expressiveness compared to other populations. The relative advantages and limitations of both interaction-based and posture-based detectors point toward the need for continued research on both types. Each type of detector captures different aspects of learners' manifestations of affective state, and many open questions remain about feature engineering and the predictive ability of each type of detector.

An important direction for future work will be the integration and combination of the two types of detectors presented here. In multiple cases, the combination of data modalities for the creation of affect detectors has been shown to produce detectors with better performance than single-modality detectors ([12], [13], [17]). As such, future work will focus on the study of how these two channels of information can be combined to produce more effective and robust detectors of affect.

Further research on effective, generalizable predictor features for posture-based affect detectors is also needed, as shown by the relatively weak predictive performance of existing features observed in this study. Complementarily, investigating the application of other machine learning algorithms, including temporal models, is likely to prove important, given the complex temporal dynamics of affect during learning. These directions are essential for developing an enhanced understanding of the interplay between affect detector architectures, learning environments, student populations, and methods for determining ground truth affect labels. While significant progress has been made toward realizing the vision of robust, generalizable affect-sensitive learning environments, these findings point toward the need for continued empirical research, as well as advances in educational data mining methods applicable to affective computing.

8. ACKNOWLEDGMENTS

We thank our research colleagues, COL James Ness and Dr. Michael Matthews in the Behavioral Science & Leadership Department at the United States Military Academy for their assistance in conducting the study. This research is supported by cooperative agreement #W911NF-13-2-0008 between the U.S. Army Research Laboratory, Teachers College Columbia University, and North Carolina State University. Any opinions, findings, and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the U.S. Army Research Laboratory.

9. REFERENCES

[1] AlZoubi, O., Calvo, R.A., and Stevens, R.H. 2009. Classification of EEG for Emotion Recognition: An Adaptive Approach. *Proc. of the 22nd Australian Joint Conference on Artificial Intelligence*, 52-61.

[2] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., and Christopherson, R. 2009. Emotion Sensors Go to School. *Proc. of the 14th Int'l Conf. on Artificial Intelligence in Education*, 17-24.

[3] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., and Rossi, L. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proc. of the 5th Int'l Conf. on Educational Data Mining*, 126-133.

[4] Baker, R., D'Mello, S., Rodrigo, M.M.T., and Graesser, A. 2010. Better to be Frustrated than Bored: The Incidence and Persistence of Affect During Interactions with Three Different Computer-Based Learning Environments. *Int'l J. of Human-Computer Studies*, 68 (4), 223-241.

[5] Baker, R.S., DeFalco, J.A., Ocumpaugh, J., and Paquette, L. 2014. Towards Detection of Engagement and Affect in a Simulation-Based Combat Medic Training Environment. *2nd Annual GIFT User Symposium (GIFTSym2).*

[6] Baker, R.S., and Ocumpaugh, J. 2015. Interaction-Based Affect Detection in Educational Software. The Oxford Handbook of Affective Computing, 233-245.

[7] Bosch, N., D'Mello, S., Baker, R., Ocumpaugh, J., Shute, V., Ventura, M., Wang, L., and Zhao, W. 2015. Automatic Detection of Learning-Centered Affective States in the Wild. *Proc. of the 2015 Int'l Conf. on Intelligent User Interfaces*.

[8] Calvo, R.A., and D'Mello, S. 2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and their

Applications. *IEEE transactions on Affective Computing*, 1 (1), 18-37.

[9] Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational Psychological Measurement*, 20 (1), 37-46.

[10] Cooper, D.G., Arroyo, I., Woolf, B.P., Muldner, K., Burleson, W., and Christopherson, R. 2009. Sensors Model Student Self Concept in the Classroom. *Proc. of the 17th Int'l Conf. on User Modeling, Adaption, and Personalization,* 30-41.

[11] D'Mello, S., and Graesser, A. 2009. Automatic detection of learners' affect from gross body language. *Applied Artificial Intelligence*, 23, 2, 123-150.

[12] D'Mello, S., Kory, J. 2012. Consistent but Modest: Comparing Multimodal and Unimodal Affect Detection Accuracies from 30 Studies. *Proc. of the 14th ACM International Conf. on Multimodal Interaction*, 31-38.

[13] D'Mello, S.K., Kory, J. in press. A Review and Meta-Analysis of Multimodal Affect Detection Systems. *ACM Computing Surveys*.

[14] D'Mello, S., Lehman, B., Pekrun, R., and Graesser, A. 2014. Confusion can be Beneficial for Learning. *Learning and Instruction*, 29, 153-170.

[15] D'Mello, S., Jackson, T., Craig, S., Morgan, B., Chipman, P., White, H., Person, N., Kort, B., el Kaliouby, R., Picard, R., and Graesser, A. 2008. AutoTutor Detects and Responds to Learners Affective and Cognitive States. *Workshop on Emotional and Cognitive Issues at the 9th Int'l Conf. on Intelligent Tutoring Systems*.

[16] Grafsgaard, J., Boyer, K., Wiebe, E., and Lester, J. 2012. Analyzing Posture and Affect in Task-Oriented Tutoring. *Proc. of the 25th Florida Artificial Intelligence Research Society Conference*, 438-443.

[17] Grafsgaard, J., Wiggins, J., Boyer, K.E., Wiebe, E., and Lester, J. 2014. Predicting Learning and Affect from Multimodal Data Streams in Task-Oriented Tutorial Dialogue. *Proc. of the 7th Int'l Conf. on Educational Data Mining*, 122-129.

[18] Hanley, J., and McNeil, B. 1982. The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143, 29-36.

[19] Litman, D.J., and Forbes-Riley K. 2006. Recognizing Student Emotions and Attitudes on the Basis of Utterances in Spoken and Tutoring Dialogue with Both Humans and Computer-Tutors. *Speech Communication*, 48, 559-590.

[20] Metcalf, S., Kamarainen, A., Tutwiler, M.S., Grotzer, T., and Dede, C. 2011. Ecosystem Science Learning via Multi-User Virtual Environments. *Int'l J. of Gaming and Computer-Mediated Simulations*, 3 (1), 86-90.

[21] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. 2006. YALE: Rapid Prototyping for Complex Data Mining Tasks. *Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*, 935-940.

[22] Nasoz, F., Alvarez, K., Lisetti, C.L., and Finkelstein, N. 2004. Emotion from Physiological Signals Using Wireless Sensors for Presence Technologies. *Cognition, Technology and Work*, 6, 4-14.

[23] Ocumpaugh, J., Baker, R.S.J.d, and Rodrigo, M.M.T. 2012. *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0.* Technical Report.

[24] Pardos, Z., Baker, R.S.J.d., San Pedro, M.O.Z., Gowda, S.M., and Gowda, S. 2014. Affective States and State Tests: Investigating how Affect and Engagement During the School Year Predict End of Year Learning Outcomes. *J. of Learning Analytics*, 1 (1), 107-128.

[25] Pekrun, R., Goetz, T., Daniels, L.M., Stupnisky, R.H., and Perry, R.H. 2010. Boredom in Achievement Settings: Exploring Control-Value Antecedents and Performance Outcomes of a Neglected Emotion. *J. of Educational Psychology*, 102, 531-549.

[26] Rai, D., Arroyo, I., Stephens, L., Lozano, C., Burleson, W., Woolf, B.P., and Beck, J.E. 2013: Repairing Deactivating Emotions with Student Progress Pages. *Proc. of the 16th Int'l Conf. on Artificial Intelligence in Education*, 795-798.

[27] Rodrigo, M.M.T., and Baker, R.S.J.d. 2009. Coarse-Grained Detection of Student Frustration in an Introductory Programming Course. *Proc. of the 5th Int'l Workshop on Computing Education Research Workshop*, 75-80.

[28] Rodrigo, M.M.T., Baker, R.S.J.d., McLaren, B., Jayme, A., and Dy, T. 2012. Development of a Workbench to Address the Educational Data Mining Bottleneck. *Proc. of the 5th Int'l Conf. on Educational Data Mining,* 152-155.

[29] Rowe, J., Mott, B., McQuiggan, J., Robison, S., and Lester, J. 2009. Crystal Island: A Narrative-Centered Learning Environment for Eighth Grade Microbiology. *Workshop on Educational Games at the 14th Int'l Conf. on Artificial Intelligence in Education*, 11-20.

[30] Ryan, R.M. 1982. Control and Information in the Intrapersonal Sphere: An Extension of Cognitive Evaluation Theory. *J. of Personality and Social Psychology*, 43, 450-461.

[31] Sanghvi, J., Castellano, G., Leite, I., Pereria, A., McOwan, P., and Paiva, A. 2011. Automatic Analysis of Affective Postures and Body Motion to Detect Engagement with a Game Companion. *Proc. of the 6th ACM/IEEE Int'l Conf. on Human-Robot Interaction*, 305-311.

[32] Sottilare, R.A., Golberg, B. and Holden, H. 2012. *The Generalized Intelligent Framework for Tutoring (GIFT).*

[33] Wixon, M., Baker, R.S.J.d., Gobert, J., Ocumpaugh, J., and Bachmann, M. 2012. WTF? Detecting Students who are Conducting Inquiry Without Thinking Fastidiously. *Proc. of the 20th Int'l Conf. on User Modeling, Adaptation and Personalization*, 286-298.

[34] Witmer, B.G., Jerome, C. J., & Singer, M. J. (2005, June). The factor structure of the presence questionnaire. *Presence*, Vol. 14(3), 298 312. MIT Press, Cambridge MA.

[35] Zeng, Z., Pantic, M., Roisman, G.I., and Huang, T.S. 2009. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (1), 39-58.

Machine beats experts: Automatic discovery of skill models for data-driven online course refinement

Noboru Matsuda[1] Tadanobu Furukawa[2] Norman Bier[3] Christos Faloutsos[2]
mazda@cs.cmu.edu tfuru@cs.cmu.edu nbier@cmu.edu christos@cs.cmu.edu
[1]Human-Computer Interaction Institute [2]Computer Science Department [3]Open Learning Initiative
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213, USA

ABSTRACT

How can we automatically determine which skills must be mastered for the successful completion of an online course? Large-scale online courses (e.g., MOOCs) often contain a broad range of contents frequently intended to be a semester's worth of materials; this breadth often makes it difficult to articulate an accurate set of skills and knowledge (i.e., a skill model, or the Q-Matrix). We have developed an innovative method to discover skill models from the data of online courses. Our method assumes that online courses have a pre-defined skill map for which skills are associated with formative assessment items embedded throughout the online course. Our method carefully exploits correlations between various parts of student performance, as well as in the text of assessment items, to build a superior statistical model that even outperforms human experts. To evaluate our method, we compare our method with existing methods (LFA) and human engineered skill models on three Open Learning Initiative (OLI) courses at Carnegie Mellon University. The results show that (1) our method outperforms human-engineered skill models, (2) skill models discovered by our method are interpretable, and (3) our method is remarkably faster than existing methods. These results suggest that our method provides a significant contribution to the evidence-based, iterative refinement of online courses with a promising scalability.

Keywords

Online course refinement, skill model discovery, evidence-base course engineering, MOOC, Q-matrix

1. INTRODUCTION

When designing and implementing large-scale online courses (aka MOOCs), defining a set of skills to be learned and having individual skills associated with particular part of course contents often becomes quite challenging. Making an effective course with explicit associations between a necessary set of skills and course contents requires intensive cognitive task analysis and time-consuming evidence-based iterative engineering [1]. Studies show how important it is to have data-analytics feedback for course improvement and theory development [2-5]. However, cognitive task analysis driven by human experts has an issue in its accuracy and scalability; applying it for a large-scale online course is often impractical.

Research shows the potential for advanced technologies to automatically and semi-automatically discover a set of skills for online courses. Learning Factor Analysis (LFA), for example, semi-automatically refines a given skill set [6]. However, LFA works only when meaningful "features" are given, which (usually) requires cognitive task analysis by subject domain experts. Other studies apply matrix factorization methods for automatic skill set (aka Q-matrix) discovery from students' response data [7, 8]. However, these methods often face the issue of interpretability—i.e., providing meaningful feedback to course designers and developers based on the machine-generated skill set is often troublesome.

We developed an efficient, practical, and scalable method that we call eEPIPHANY, to fully and automatically discover skill sets from online course data, which are the combination of the assessment item text data (i.e., problem and feedback text sentences for assessment items) and student learning interaction data. eEPIPHANY is a collection of data-mining techniques to automatically refine (or rebuild) a human-crafted set of skills, initially given by course designers and developers.

The most important goal of eEPIPHANY is to provide constructive feedback to online course designers and developers for iterative course improvement. We assume that our target online courses have occasional formative assessments to probe students' competency towards learning objectives. We hypothesize that students' response data and assessment item text data both reflect latent skills to be learned, and assessment items can be clustered based on those latent skills. To test these hypotheses, we implemented eEPIPANY as a combination of the matrix factorization to analyze students' response data and bag-of-words techniques to analyze course content data.

The contributions of this work are the following: (1) *A new problem formulation*—We show how to integrate diversified information such as student performance and assessment item text data. (2) *A new algorithm*—Our solution, the eEPIPHANY algorithm, is scalable and effective for practical use for large-scale online course engineering. (3) *Evaluation*—eEPIPHANY outperforms past competitors, including *human experts*, on several, real online course datasets.

The goal of this paper is to introduce the eEPIPHANY method (section 3) and provide empirical evaluation for its effectiveness (section 4). We discuss implications for the application of eEPIPHANY to evidence-based online course refinement (section 5.3). To begin, the next section provides a standard structure of our target online courses and various definitions for later discussions.

2. SKILL MODEL FOR ONLINE COURSES

We assume that our target online courses have occasional low-stake assessments throughout the course—aka formative assessments—to assess students' competency on target *skills*. We assume that each formative assessment has multiple *assessment items* (i.e., problems to answer), each of which is associated with one or more skills.

We assume that online courses have a pre-defined *skill map* (often called Q-matrix [9, 10]) that shows one-to-many mapping between individual skills and one or more assessment items. In this paper a mapping between a single skill and multiple assessment items in the skill map is called a *skill-item association*.

We call a set of skills a *skill model*. The terms "skill model" and "skill map" will be used interchangeably in this paper. The pre-defined skill model is therefore called the "*default*" *skill model*—a human-developed model that is initially guided by authors' intuition in the absence of data, or a human-developed model that has been refined based on student data.

The Open Learning Initiative (OLI) at Carnegie Mellon University [11] is an example of an online course platform that meets the above-mentioned criteria [12]. OLI courses all have a *human-crafted* "default" skill model that is often recognized as semi-optimal, and could always be improved.

To improve skill models to refine online courses, it becomes crucial that the machine-discovered skill models have high interpretability so that course designers and developers can make sense of the proposed skill model improvements. Our proposed method, eEPHIPHANY, discovers accurate and interpretable skill models from learning data and assessment item text data. The next section describes details of the eEPHIPHANY method.

3. eEPIPHANY

eEPIPHANY is a collection of data mining techniques for automatic discovery of skill models from online course data. The primary input to eEPIPHANY is a matrix representing a chronological record of students' responses to assessment items, called an A-matrix (Figure 6-a). The A-matrix is a three-dimensional matrix showing a history of attempts on individual assessment items made by individual students. Each attempt is a vector of binary values representing the correctness of a student's response—0 indicates incorrect and 1 indicates correct. The A-matrix contains at most one correct response per student per assessment item.

The goal of eEPIPHANY is to find a skill model (Q-matrix) that produces the best prediction of the A-matrix. The predictive power is measured by cross-validation. eEPIPHANY can either find a Q-matrix by itself or refine a given Q-matrix by the following steps: (1) clustering assessment items with latent features that would best characterize the similarity in the difficulties of assessment items (section 3.1), (2) proposing a new skill model by assuming that the above-mentioned cluster of assessment items provides a hint for new skills (section 3.2), and (3) searching for the best skill model by comparing multiple skill model candidates (section 3.3).

3.1 Feature Extraction

We have developed two latent-feature extraction strategies: (1) the Matrix Factorization (MF) strategy, and (2) the Bag-of-Words (BoW) strategy. The goal of feature extraction, regardless of the strategy difference, is to generate a two-dimensional matrix, the P-Matrix, showing a mapping between assessment items and "skill candidates" (Figure 6-d. Also see below).

3.1.1 Matrix factorization (MF) strategy

For the MF strategy, the A-matrix is first transformed into the difficulty matrix (D-matrix), which is a two-dimensional matrix representing an individual student's difficulty for each assessment item. We hypothesize that the record of individual students' performance on assessment items reflect their "difficulties" in answering assessment items, and that those students who show a similar distribution pattern of difficulties share a similar competency on latent skills.

The item difficulty *id*, by definition, is computed as $id = 1 - 1/d$ where d is the number of attempts made on an assessment item. We only include attempts until the first correct attempt is made, i.e., *id* is the length of the vector of attempts in the A-matrix (Figure 6-a). We hypothesize that students would more likely skip items that look too easy for them hence no difficulties at all. Therefore, we defined *id* as 0 for missing data in the A-Matrix (i.e., skipped items).

The D-matrix is then factorized into U and V matrices (i.e., $D = U \times V$) by the Non-Negative Matrix Factorization method [13]. The V-matrix is a two-dimensional (assessment item by latent feature) matrix. It is therefore a collection of *feature vectors*, each corresponding to an assessment item (Figure 6-b).

Assessment items in the V-matrix are then clustered by the k-means method [14], resulting in an F-matrix (Figure 6-c). We hypothesize that each cluster in the F-matrix represents a "skill candidate" that can be used to construct the P-Matrix (Figure 6-d).

The P-Matrix is a two-dimensional binary matrix showing which assessment item belongs to which skill candidate. The P-matrix represents the association of each assessment item to a skill candidate. By its nature, in the current eEPIPHANY algorithm, each assessment item has an association to at most one skill candidate (if any).

3.1.2 Bag-of-words (BoW) strategy

The BoW strategy creates the F-matrix directly from a collection of *item stems* (i.e., assessment item text data showing problem and feedback texts) for assessment items. That is, the assessment items are clustered by the bag-of-words method using item stems.

We first transform each assessment item into a set of component words from a collection of item stems using a part-of-speech tagger, TreeTagger[1]. We then apply the Latent Dirichlet Allocation model (LDA) [15] to cluster assessment items. Assessment items are clustered based on the probability of topic distribution—i.e., individual assessment items are assigned to the topic with the highest topic probability, which results into the F-Matrix from which the P-Matrix is generated as mentioned above.

[1] www.cis.uni-muenchen.de/~schmid/tools/TreeTagger

3.2 Skill Model Construction

eEPIPHANY refines a given "default" skill model by either modifying it or replacing it with a new skill model. In either case, eEPIPHANY first proposes candidates for new skills, and then finds the best way to refine the default skill model in terms of the accuracy of the data fit. This subsection describes the former step, whereas the latter step is described in section 3.3.

Given a P-matrix, there are three strategies to refine the "default" skill model: (1) Replacing the entire "default" skill model with an entirely new skill model, (2) appending new skill-item associations to the "default" skill model, (3) splitting given a skill-item association(s) in the "default" skill model into multiple skill-item associations.

3.2.1 Replace Strategy

To replace the default skill model with an entirely new skill model, the P-matrix is straightforwardly converted into the Q-matrix. Namely, each skill candidate becomes a new skill. Assessment items that are associated with the skill candidate become members of the skill-item association for the newly defined skill.

3.2.2 Append Strategy

The *append* strategy adds more skill-item associations to the default skill model, while the original skill-item associations in the default skill model remain intact. Skill-item associations that are being newly added are the same set of skill-item associations proposed by the *replace* strategy. The following example illustrates this process (Figure 1):

Assume that there is a skill-item association a_i for a skill s_i with assessment items $q^i_1 \ldots q^i_5$ in the default skill model. Also, assume that there is a skill candidate c_1 and c_2 in the P-matrix where c_1 has a skill-item association with assessment items q^i_1, q^i_2, and q^i_3; and c_2 has a skill-item association with assessment items q^i_4 and q^i_5. The *append* strategy enters two new skill-item associations, one for c_1 and another one for c_2 into the default skill model. As a consequence, the assessment item q^i_1, for example, is now associated with two skills, s_i and c_1.

It is worth noting that the skill model produced by the *replace* strategy is the proper subset of the skill model produced by the *append* strategy. The number of skills in the skill model produced by the *append* strategy is the sum of the number of skills in the default skill model and the number of skills in the skill model produced by the *replace* strategy.

3.2.3 Split Strategy

The *split* strategy refines the default skill model by individually splitting skill-item associations into multiple new skill-item associations. These splits are based on skill-item associations in

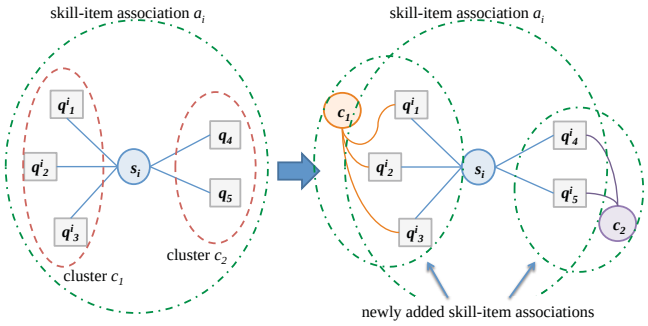


Figure 1. The *append* strategy appends new skill-item associations to the default skill model

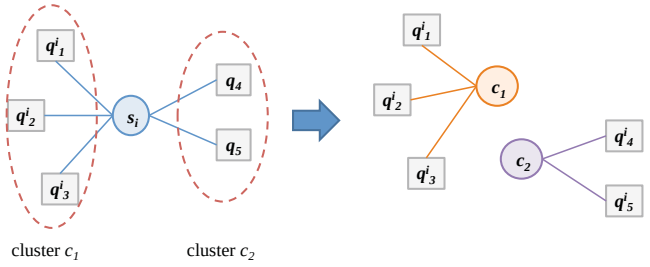


Figure 2. The *split* strategy breaks given skill-item associations into new ones with newly discovered skills

the P-Matrix. The following example illustrates this process (Figure 2):

Assume the same situation as mentioned above for the *append* strategy. That is, there is a skill-item association a_i for a skill s_i with assessment items $q^i_1 \ldots q^i_5$ in the default skill model. Also, assume that there is a skill candidate c_1 and c_2 in the P-matrix where c_1 has a skill-item association with q^i_1, q^i_2, and q^i_3; and c_2 has a skill-item association with q^i_4 and q^i_5. The *split* strategy then replaces the original skill-item association a_i with two new skill associations a_{i-1} and a_{i-2}, where a_{i-1} has c_1 as a skill and q^i_1, q^i_2, and q^i_3 as associated assessment-items, while a_{i-2} has c_2 as a skill and q^i_4 and q^i_5 as associated assessment-items.

3.3 Model Search

We hypothesize that two different types of feature-extraction strategies (section 3.1) present pros and cons for our purposes. For example, the item stem (i.e., problem sentences and feedback messages) might reflect skills necessary to answer the assessment item correctly. On the other hand, the student response data might reflect skills that students have actually acquired. The BoW strategy might provide better interpretability, but the student response data might provide more accurate skill models. The BoW strategy can be applied even before the course has been used (i.e., before student data is available).

With the lack of a predictive theory of parameter selection to compute the best skill model, eEPIPHANY exhaustively searches for the best skill model by comparing all possible skill models with different combinations of the following four parameters. The comparison is done by the model fit using the Bayesian Knowledge Tracing as a predictor:

(1) The number of components used for the Matrix Factorization (N_C)—This determines a dimension of the V-matrix. N_C reflects the variance in the pattern of student competency over the latent features. Although, the greater N_C value would result in the smaller reconstruction error (i.e., ||D-U*V||), it might also result in the over fit to the data (which is penalized in the AIC and BIC scores). *Nc* varies from 10 to the number of students, increased by 10 during the model search.

(2) The number of clusters in k-means (N_k)—We hypothesize that each feature is shared by at least five assessment items. Therefore, N_k varies from 25 to $N_Q/5$ where N_Q is the number of assessment items; increased by 25 during the model search.

(3) The number of topics used for LDA (section 3.1.2) to compute the bag-of-words clustering (N_T)—Here again, applying the same hypotheses as for N_k. N_T varies from 25 to $N_Q/5$, increased by 25 during the model search.

(4) The threshold used for the split strategy (β)—Assume that skill s is associated with n assessment items, $q_i,..,q_n$. Also assume

that in the P-matrix, these n assessment items are associated with k skill candidates, $C = \langle c_1,\ldots,c_k \rangle$. The skill-item association for s will be split into new skill-item associations with skill candidate c in C, if the number of assessment items associate with the skill candidate c is greater than $n \times \beta$. β is set to 0.05, 0.25, and 0.5 in this order during the model search.

3.4 Model Interpretation: The DoE Analysis

The most important goal of the skill-model discovery and refinement proposed in the current paper is to improve online courses. Providing ***interpretable*** feedback based on a machine-discovered skill model and model refinement is therefore crucial. We hypothesize that to achieve this goal, two subgoals must be met: (1) to identify what part of the default skill model has been improved the most, and (2) to understand the improvement from a domain perspective.

To identify the part of the skill model that has been improved most, we analyze the *degree of enhancement* (DoE) of the proposed change in skill models. We hypothesize that the DoE would be maximized among a skill(s) for which the accuracy of students' performance prediction improved the most [16]. The accuracy of student performance prediction is operationalized as the root mean squared error (RMSE) in cross-validation for the model-fit evaluation.

Based on this hypothesis, we identify skills with the most DoE in the default skill model M_D relative to a refined (i.e., machine-discovered) skill model M_R as follows:

(1) For each skill s_i in the default skill model M_D, let I_D^i be a set of assessment items associated with s_i.

(2) Find all skills c_i^j (j=1,…,n_i) in the refined skill model M_R that are associated with any assessment items in I_D^i.

(3) Compute xI_D^i, the extended version of I_D^i, by adding all assessment items associated with any of c_i^j to I_D^i.

(4) Compute RMSEs_i that is an RMSE in predicting student performance on assessment items in xI_{Di} using corresponding s_i in M_D as the predictor.

(5) Compute RMSEc_i that is an RMSE in predicting student performance on assessment items in xI_{Di} using corresponding c_i^j in M_R as a predictor.

(6) Let d_i = RMSEs_i – RMSEc_i be the *DoE score* of skill s_i relative to c_i^j.

(7) Find a skill s in M_D with the largest DoE score. The skill s has the largest error reduction from M_D to M_R.

Once the skill with the largest error reduction is found, the next step is to understand what the improvement is about, that is, to interpret the machine-discovered model refinement with the focus on the skill with the largest error reduction.

To interpret the proposed model refinement, we use the bag-of-words analysis in combination with manual inspection of the assessment item text. For each skill-item association in the refined skill model, a set of keywords is extracted from the item stem (i.e., the combination of text sentences for the items and their feedback messages). The χ^2 value is computed for individual word w appearing in the item stem for a skill-item association k as follows [17]: $\chi^2(k, w) = (\text{aic}(k, w) - \text{aict}(k, w))^2 / \text{aict}(k, w)$ where $\text{aic}(k, w)$ is the number of assessment items that contains w in k, and $\text{aict}(k, w)$ is a theoretical implication for $\text{aic}(k, w)$, i.e., $\text{aict}(k, w) = \text{aic}(k, *) \times \text{aic}(*, w) / \text{aic}(*,*)$. The word w is considered as a keyword only when $\text{aict}(k, w) < \text{aic}(k, w)$.

Table 1. Three OLI datasets used for the evaluation

	Statistics	Biology	C@CM
#Students	1,013	481	100
#Transactions	538,062	418,344	94,612
#Unique Items	1,791	916	912

4. EVALUATION

To evaluate the efficiency and effectiveness of the eEPIPHANY method, we applied it to actual online course data.

4.1 Data

Three OLI courses—Computing@CarnegieMellon (C@CM), Biology, and Statistics—were used for evaluation. All three courses are actively used at Carnegie Mellon University and other educational institutions for registered, academic students and in open sections for independent learners. Table 1 shows the number of students, transactions (i.e., students' responses to assessment items), and unique items; these datasets represent use in academic contexts. All these OLI data are available on DataShop [18]. It turned out that the C@CM data only contains randomly selected students' data from a larger pool of the OLI data that contains more than 1300 academic students enrolled.

4.2 Method

For each of the three OLI datasets, we applied eEPIPHANY and had it search the best skill model by finding the optimal clustering parameters (section 3.3). During the search we recorded the model-fit for three feature-extraction strategies (matrix factorization, bag-of-words, and their combination as described in section 3.1) crossed over three skill-model construction strategies (*split*, *add*, and *replace* as in section 3.2). The model-fit was computing by cross-validation using the Bayesian Knowledge Tracing technique.

4.3 Results

4.3.1 Comparison of feature extraction and refinement strategies

Table 2 shows the best skill models, annotated with the strategies and parameters used to discover them. As the table shows, *the matrix factorization (MF) strategy always outperformed the BoW strategy for the three datasets used in the study.* When the MF strategy is used, *replacing the default skill model with a completely new skill model discovered by eEPIPHANY yielded the best skill model* for all dataset.

To understand how the size of cluster impacts the quality of the resultant skill model, we compared different skill models with different sizes measured as the number of skills. Figure 3 plots the

Table 2. ePIPHANY always found better skill model than experts. FS: Feature Extraction Strategy, SC: Skill Construction Strategy, #S: Number of items

FS	SC	#S	AIC	BIC	RMSE
Statistics					
MF	Replace	63	307730	310731	0.447
BoW	Append	143	317808	323802	0.456
Biology					
MF	Replace	86	224944	228514	0.389
BoW	Split	187	228597	236360	0.393
C@CM					
MF	Replace	41	59497	60998	0.364
BoW	Split	137	61648	66661	0.371

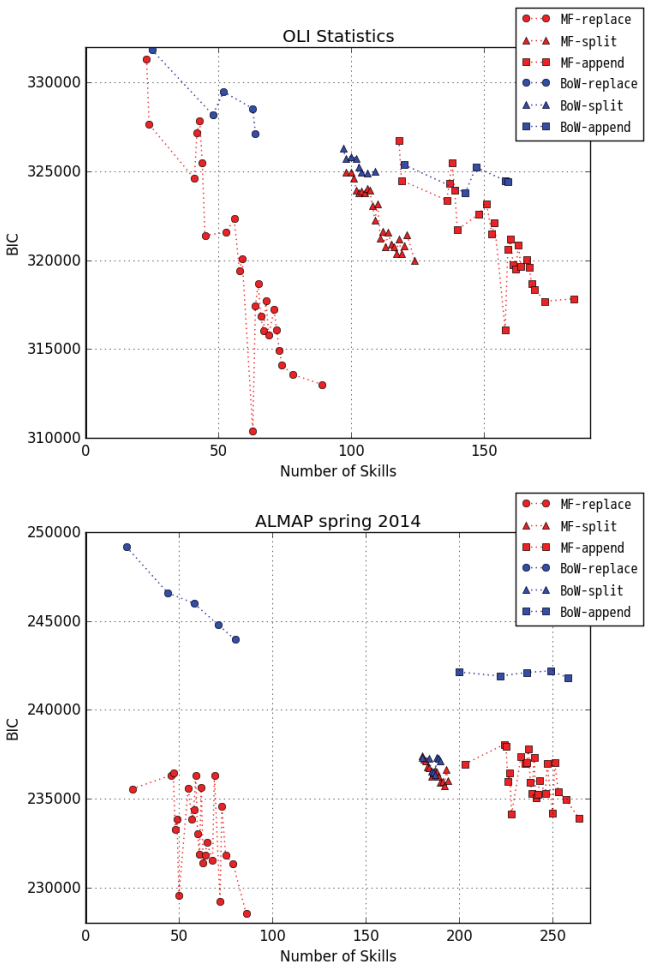


Figure 3. MF-replace wins or ties with MF-append: Comparison of skill models with different size. OLI Statistics (top) and Biology (bottom)

BIC (Y-axis) against a number of skills (X-axis). In the figure, two feature extraction strategies—MF and BoW—are crossed three skill-model construction strategies—replace, split, and append.

As the figure shows, it turned out that *for any strategy combination, the bigger the size of the model (i.e., the number of the clusters) the better the model.* It can be also seen that the *replace strategy is relatively better than other two skill-model construction strategies* (as depicted by more dots towards the bottom).

4.3.2 Comparison with other methods

Table 3 shows the comparison of the model-fit between skill models discovered by LFA, an OLI course designer (OLI), and eEPIPHANY (eEPI) on the OLI Statistics course. In DataShop, skill models discovered by LFA and human expert only contain data from Unit 1. Therefore, for this analysis, we applied eEPIPHANY only to the OLI data from Unit 1.

The table shows the number of skills (#S) and the number of assessment items (Obs.). The model fit was evaluated by AIC, BIC, and RMSE scores computed by using Additive Factor Model (AFM) [19] and Bayesian Knowledge Tracing (BKT). As shown in the table, eEPHIPHANY outperformed human expert (OLI),

Table 3. eEPIPHANY beats human expert on OLI Statistics. The analysis contains data only from Unit 1.

Method	#S	Obs.	AIC	BIC	RMSE
AFM					
eEPI	22	75955	72125	80901	0.412
LFA	28	75955	69108	77984	0.404
OLI	19	75955	74787	83507	0.418
BKT					
eEPI	22	75955	74560	75373	0.407
LFA	28	75955	74343	75378	0.404
OLI	19	75955	77405	78107	0.414

Table 4. Assessment items involved in the most beneficial skill model refinement

ID(Skill)	Assessment item (item stem)
Q881(c31)	The ability or tendency of organisms and cells to maintain stable internal conditions is called homeostasis (value:A) metabolism (value:B) evolution (value:C) emergent property (value:D)
Q885(c31)	Why do organisms maintain fairly steady conditions within their cells and bodies? They need to keep conditions stable so that they can obtain food. (value: A) Organisms just change along with whatever is happening in the outside world, which is usually quite steady. (value: B) They must maintain stable conditions to keep their enzymes working and generally to enable the chemical reactions of life. (value: C) Unstable conditions will destroy the DNA in cells; this is the most important risk for a cell facing physical or chemical stress. (value: D)
Q901(c31)	An organism or cell exhibits _____ when it maintains steady internal conditions despite changes in the outer environment. homeostasis (value: A) evolution (value: B) natural selection (value: C) balance (value: D)
Q717(c3)	Humans maintain a blood pH between 7.35 and 7.45. In order to maintain homeostasis, how will your body respond if your blood pH drops to 7.0? If your blood pH is 7.0, your body will raise your pH. (value: A) If your blood pH is 7.0, your body will lower your pH. (value: B) A blood pH of 7.0 is close enough to 7.35. Your body won't do anything. (value: C)

and arguably tied with LFA. We will further discuss this result in section 5.3.

4.3.3 Model interpretation

Figure 5 shows the skill *k153* with the largest DoE score (section 3.4) in the OLI Biology course. In the figure, the skill *k153* in the default skill model was associated with four assessment items. In the discovered skill model, these 4 assessment items are associated with two skills—*c31* and *c3*. The newly constructed skills *c31* and *c3* have 16 and 19 assessment items associated respectively. The RMSE is computed for those 35 steps using skills in the default skill model. The RMSE is then re-computed using *c31* and *c3*. According the DoE analysis, splitting skill *k153* into two skills *c3* and *c31* yields the biggest DoE score. This addressed the first subgoal of the model interpretation.

To interpret model improvement, we investigated four assessment items associated with *k153* in the default skill model to see why they were split into two groups. Table 4 shows four assessment

Table 5. Bag of words for a skill (k153) split into two new skills (c31 and c3)

Skill	Bag of Words
k153	homeostasis range internal maintain steady condition narrow tendency metabolism raise optimal entitiy exhibit sensitive balance chemistry drop world despite happening
c31	steady homeostasis evolutionary stress valid theme progress favor module tree ancestor selection adapt internal evolution ancestry natural conclusion environmental whale
c3	hazy fundamentally matter space play concept structo yet mass nutrient exchange determine sometimes dramatically biology rule ability quite period peanut

items and their skill association in the refined skill model. Table 5 shows the bag-of-words associated with each skill cluster.

In the default skill model, the skill *k153* is to "Define homeostasis and explain its role in maintaining life." All four assessment-items related to *k153* in the default skill model mention "homeostasis" and "sustainable life." However, a closer look shows that this skill is most appropriate for the three out of four assessment items—Q881, Q885, and Q901. In the refined skill model, these three assessment items are correctly tagged as one skill *c31*.

Although the fourth assessment item Q717 relates to homeostasis, a closer look shows that learners are being asked to engage in a more sophisticated task—i.e., determine (or predict) necessary action to achieve homeostasis, which results in a separate association with skill *c3*.

For those four rows, the machine-generated split is very coherent from a subject-matter expert's perspective. This satisfies the second subgoal of the model interpretation.

4.3.4 Efficiency

One of the notable strengths of the eEPIPHANY method is its efficiency. As described in section 3.3, eEPIPHANY searches the best skill model by a brute-force search by merely changing the number of clusters, which takes linear time O(*n*). This linear computation must be repeated nine times for three different feature-extraction strategies crossed with three different skill-model construction strategies, which still takes O(n).

The Learning Factor Analysis (LFA) method [6] requires an intensive search for each skill (*s*) over multiple difficulty factors (*d*) that takes O(s^d).

During the evaluation study that used three real OLI course data, eEPIPHANY found the best model in 2 to 3 hours per dataset running on a single-core personal computer, showing its practical potential for actual application to large-scale online course improvement.

5. DISCUSSION

5.1 Strategy comparison

Our study showed that using student response data (i.e., the number of attempts made on assessment items before a student finally made their first correct response) always yields a better skill model than using the bag-of-words with item stems. We also found that *even only using the bag-of-words, eEPIPHANY always yields a better skill model than the default skill model that is hand-crafted by human experts*.

As for the skill-model construction strategy, the *replace* strategy always discovers the best skill model in our study, suggesting that *the Matrix Factorization strategy efficiently discovers a latent skill model from the student learning data*. On the other hand, the *split* strategy always resulted in producing an inferior skill model in our study; suggesting that *the* split *strategy hardly improves on the human-crafted skill*.

The above observation also implies that *eEPIPHANY can actually find a better skill model completely automatically without human interaction (which is what the replace strategy does) from real online course data*.

5.2 Interpretability

To interpret skill models proposed by the Matrix Factorization (MF) strategy is to interpret clusters of assessment items, which is often quite challenging. For the purpose of course refinement however, interpretability becomes crucial.

To overcome this issue, while still taking the advantage of the MF strategy to produce high-quality skill models, we applied the degree of enhancement (DoE) analysis to identify the instance of refinement that received the most benefit—i.e., identifying the skill that received the largest benefit from skill decomposition. We also combined the bag-of-words technique with manual inspection. Our study demonstrated that *this hybrid technique allows course designers to make meaningful interpretations of the proposed refinements of the skill model*.

Yet the obvious limitation of the current technique is its dependence on manual inspection. We hypothesize that one idea to overcome this issue is to combine MF and BoW, namely, to expand the V-matrix (Figure 6-b) by adding the bag-of-words keyword information as a latent feature, and then applying k-mean clustering. The resulting clusters (i.e., the skill candidates) would have better interpretability supported by the bag-of-words keyword information. Testing this hypothesis is an important future study.

5.3 Implication for evidence-based online course refinement

Our study demonstrated that eEPIPHANY discovers skill models that reflect student learning more accurately than human-crafted skill models on all three OLI course data. Even though eEPIPHANY requires human labor to interpret the discovered skill models (with the aid of DoE), it is arguably still less time consuming than creating skill models by hand. Figure 4 depicts this argument as a two-dimensional plot.

We also argue that eEPIPHANY is less labor intensive than LFA, because LFA requires human experts to generate the P-Matrix, which usually requires time-consuming cognitive task analysis. The high demand on human labor might not practical and hence might not scale up to apply to large online courses such as OLI. In fact, as far as we know, there has been no actual application of LFA with human-crafted P-Matrix to OLI courses. In the comparison in Table 3, the data for LFA is taken from DataShop [18], but LFA for these skill models used other existing skill models as P-Matrix (personal communication), therefore, it is not actually a fair comparison—LFA shows in this paper does not use the P-Matrix created by human experts. On the other hand, eEPHIPHANY automatically discover the P-Matrix from data.

Nonetheless, as our study has shown, eEPIPHANY and LFA discovered equally accurate skill models. We also found that different evaluation criteria (i.e., AFM vs. BKT in Table 3) show different favors on different search algorithm. LFA uses AFM and ePIPHANY uses BKT as a search bias, and that might have affected the results. We have yet to investigate this issue.

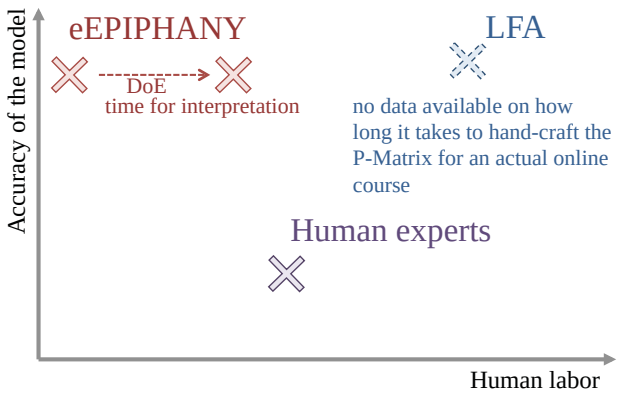


Figure 4. eEPIPHANY discovers skill models better than human experts and quicker than LFA

For our core goal—to provide evidence-based feedback for online course refinement—our study also suggests that eEPIPHANY can be used for a dual purposes with regard to skill model improvement: (1) When the online course is initially implemented, we should apply eEPIPHANY with the bag-of-words strategy. (2) When the online course is actually used and student learning data are collected, then we should apply eEPIPHANY with the student data to further improve the course.

The above observations further suggest that *authors of online courses would not need to create a default skill model at all—eEPIPHANY can find the default model by itself using the bag-of-words method*. This rather strong argument must be investigated as future research.

6. CONCLUSION

We found that eEPIPHANY is an efficient, practical, and quick method to automatically discover skill models from online course data without human interaction. Our empirical study showed that eEPIPHANY always finds skill models that are better than human-crafted skill models used in actual online courses. We also demonstrated that eEPIPHANY-crafted skill models have reasonable interpretability with the added help of the text analysis technique.

Creating effective online courses often requires intensive, iterative system engineering. Studying techniques for automatic skill model refinement and its application for evidence-based course refinement therefore is a critical research agenda for the successful future of online education.

ACKNOWLEDGEMENT

The research reported here was supported by National Science Foundation Awards No.1418244.

7. REFERENCES

1. Fishman, B., et al., *Creating a Framework for Research on Systemic Technology Innovations.* The Journal of the Learning Sciences, 2004. **13**(1): p. 43-76.
2. Stamper, J.C. and K.R. Koedinger, *Human-machine student model discovery and improvement using data*, in *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, S.B. G. Biswas, J. Kay, & A. Mitrovic, Editor. 2011, Springer: Berlin. p. 353-360.
3. Koedinger, K.R., et al., *Using Data-Driven Discovery of Better Student Models to Improve Student Learning*, in *Proceedings of the International Conference on Artificial Intelligence in Education*, H.C. Lane, et al., Editors. 2013, Springer: Memphis, TN. p. 421-430.
4. Velmahos, G.C., et al., *Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory.* Am. J. Surg, 2004. **18**: p. 114–119.
5. Koedinger, K.R. and E.A. McLaughlin, *Seeing language learning inside the math: Cognitive analysis yields transfer*, in *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, S. Ohlsson and R. Catrambone, Editors. 2010, Cognitive Science Society: Austin, TX.
6. Cen, H., K. Koedinger, and B. Junker, *Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement*, in *Intelligent Tutoring Systems*, M. Ikeda, K. Ashley, and T.-W. Chan, Editors. 2006, Springer Berlin Heidelberg. p. 164-175.
7. Desmarais, M.C., *Mapping Question Items to Skills with Non-negative Matrix Factorization.* SIGKDD Explor. Newsl., 2012. **13**(2): p. 30--36.
8. Sun, Y., et al., *Alternating Recursive Method for Q-matrix Learning*, in *Proceedigns of the International Conference on Educational Data Mining*, J. Stamper, et al., Editors. 2014. p. 14-20.
9. Barnes, T., *The Q-matrix Method: Mining Student Response Data for Knowledge*, in *Proceedings of AAAI 2005 Educational Data Mining Workshop*. 2005.
10. Tatsuoka, C., et al., *Developing Workable Attributes for Psychometric Models Based on the Q-Matrix.* Journal for Research in Mathematics Education, accepted.
11. Lovett, M., O. Meyer, and C. Thille, *The Open Learning Initiative: Measuring the effectiveness of the OLI statistics course in accelerating student learning.* Journal of Interactive Media in Education, 2008.
12. Bier, N., R. Strader, and D. Zimmaro, *An Approach to Skill Mapping in Online Courses*, in *Learning with MOOCs*2014: Cambridge, MA.
13. Lee, D.D. and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization.* Nature, 1999. **401**.
14. MacQueen, J., *Some methods for classification and analysis of multivariate observations*, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967.
15. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent Dirichlet Allocation.* The Journal of Machine Learning Research, 2003. **3**.
16. Koedinger, K.R., E.A. McLaughlin, and J.C. Stamper, *Automated student model improvement*, in *Proceedings of the 5th International Conference on Educational Data Mining*, K. Yacef, et al., Editors. 2012. p. 17-24.
17. Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. 2008, New York, NY: Cambridge University Press.
18. Koedinger, K.R., et al., *A Data Repository for the EDM community: The PSLC DataShop*, in *Handbook of Educational Data Mining*, C. Romero, et al., Editors. 2010, CRC Press: Boca Raton, FL.
19. Cen, H., K.R. Koedinger, and B. Junker, *Is over practice necessary? – improving learning efficiency with the Cognitive Tutor through educational data mining*, in *Proceedings of 13th International Conference on Artificial Intelligence in Education*, R. Luckin, K.R. Koedinger, and J. Greer, Editors. 2007, IOS Press: Amsterdam. p. 511-- - 518.

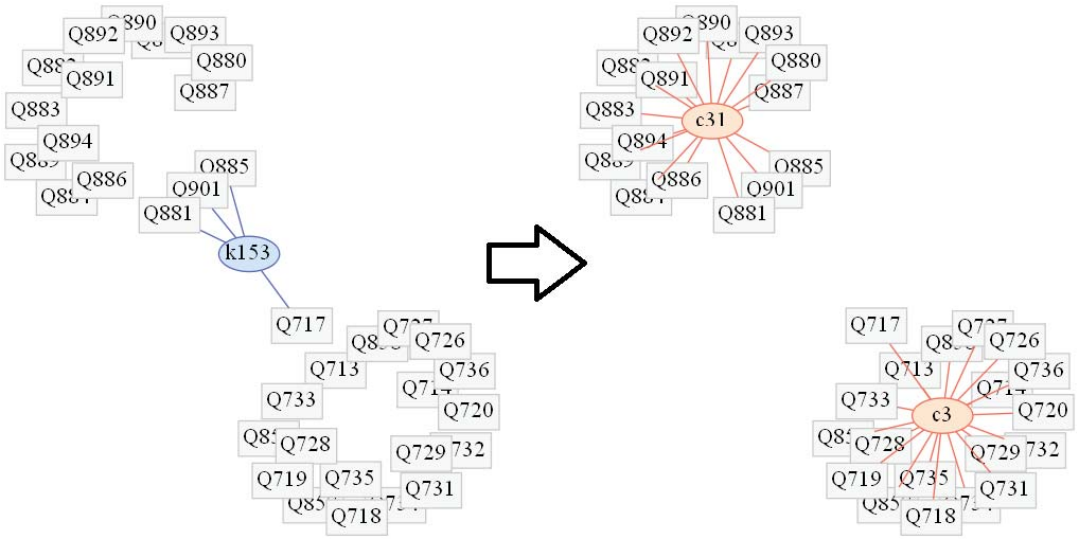


Figure 5. eEPIPHANY agrees with intuition: Assessment items are plotted in a skill-item association. (a) In the default skill model (left), skill *k153* are associated with assessment items Q881, Q885, and Q901 in the default skill model). (b) In the refined skill model (right), these three assessment items are associated with two skills (*c3* and *c31*) among others. In the figure, those other skills plotted in the "default" skill model are the ones contained in xI_D^I (section 3.4).

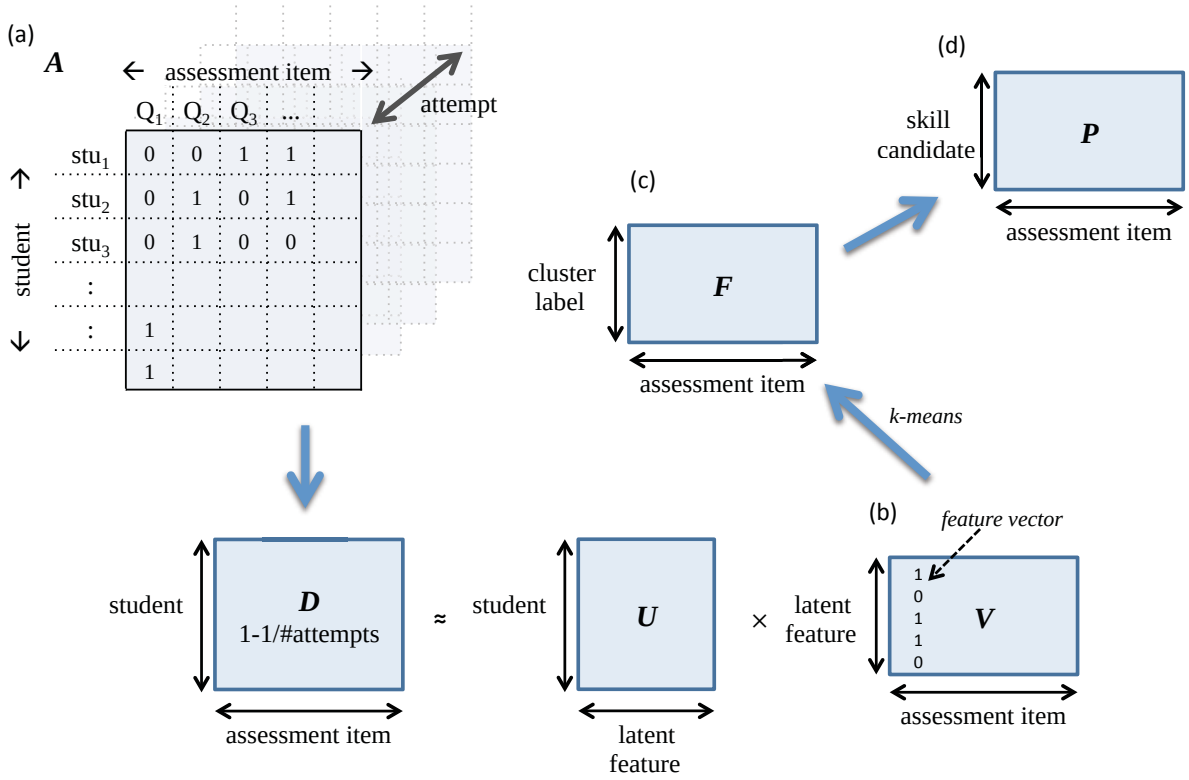


Figure 6. Overview of eEPIPHANY

Student Models for Prior Knowledge Estimation

Juraj Nižnan
Masaryk University Brno
niznan@mail.muni.cz

Radek Pelánek
Masaryk University Brno
xpelanek@mail.muni.cz

Jiří Řihák
Masaryk University Brno
thran@mail.muni.cz

ABSTRACT

Intelligent behavior of adaptive educational systems is based on student models. Most research in student modeling focuses on student learning (acquisition of skills). We focus on prior knowledge, which gets much less attention in modeling and yet can be highly varied and have important consequences for the use of educational systems. We describe several models for prior knowledge estimation – the Elo rating system, its Bayesian extension, a hierarchical model, and a networked model (multivariate Elo). We evaluate their performance on data from application for learning geography, which is a typical case with highly varied prior knowledge. The result show that the basic Elo rating system provides good prediction accuracy. More complex models do improve predictions, but only slightly and their main purpose is in additional information about students and a domain.

1. INTRODUCTION

Computerized adaptive practice [14, 22] aims at providing students with practice in an adaptive way according to their skill, i.e., to provide students with tasks that are most useful to them. In this work we focus on the development of adaptive systems for learning of facts, particularly on modeling of prior knowledge of facts.

In student modeling [6] most attention is usually paid to modeling student learning (using models like Bayesian Knowledge Tracing [4] or Performance Factors Analysis [24]). Modeling of prior knowledge was also studied in prior work [22, 23], but it gets relatively little attention. It is, however, very important, particularly in areas where students are expected to have nontrivial and highly varying prior knowledge, e.g., in domains like geography, biology, human anatomy, or foreign language vocabulary. As a specific case study we use application for learning geography, which we developed in previous work [22]. The estimate of prior knowledge is used in models of current knowledge (learning), i.e., it has important impact on the ability of the practice system to ask suitable questions.

We consider several approaches to modeling prior knowledge and explore their trade-offs. The basic approach (described in previous work [22]) is based on a simplifying assumption of homogeneity among students and items. The model uses a global skill for students and a difficulty parameter for items; the prior knowledge of a student for a particular item is simply the difference between skill and difficulty. The model is basically the Rasch model, where the parameter fitting is done using a variant of the Elo rating system [9, 25] in order to be applicable in an online system.

The first extension is to capture the uncertainty in parameter estimates (student skill, item difficulty) by using Bayesian modeling. We propose and evaluate a particle based method for parameter estimation of the model. This approach is further extended to include multiplicative factors (as in collaborative filtering [15]) which allows to better model the heterogeneity among students and items.

The second extension is the hierarchical model which tries to capture more nuances of the domain by dividing items into disjoint subsets called concepts (or knowledge components). The model then computes student skill for each of these concepts. Since these concept skills are related, they are still connected by a global skill. With this model we have to choose an appropriate granularity of used concepts and find an assignment of items to these concepts. We use both manually determined concepts (e.g., "continents" in the case of geography) and concepts learned automatically from the data [19].

The third extension is a networked model, which bypasses the choice of concepts by modeling relations directly on the level of items. This model can be seen as a variation on previously proposed multivariate Elo system [7]. For each item we compute the most similar items (based on students' answers), e.g., in the geography application, knowledge of Northern European countries is correlated. Prior knowledge of a student for a particular item is in this model estimated based on previous answers to similar items (still using the global skill to some degree).

Extended models are more detailed than the basic model and can potentially capture student knowledge more faithfully. They, however, contain more parameters and the parameter estimation is more susceptible to the noise in data. We compare the described models and analyze their performance on a large data set from application for learning geography [22].

The results show that the studied extensions do bring an improvement in predictive accuracy, but the basic Elo system is surprisingly good. The main point of extension is thus in their additional parameters, which bring an insight into the studied domain. We provide several specific examples of such insight.

2. MODELS

Although our focus is on modeling knowledge of facts, in the description of models we use the common general terminology used in student modeling, particularly the notions of *items* and *skills*. In the context of geography application (used for evaluation) items correspond to locations and names of places and skill corresponds to knowledge of these facts.

Our aim is to estimate the probability that a student s knows an item i based on previous answers of students s to questions about different items and previous answers of other students to questions about item i. As a simplification we use only the first answer about each item for each student.

In all models we use the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ as a link between a skill and a probability that a student answers correctly. In the case of multiple-choice questions the probability can be modeled by a shifted logistic function $\sigma(x,k) = 1/k + (1-1/k)\frac{1}{1+e^{-x}}$, where k is the number of options. We restrict our attention to online models (models that are updated after each answer). Such models can adapt to user behavior quickly and therefore are very useful in adaptive practice systems.

2.1 Basic Model

The basic model (described in previous work [22] and currently used in the online application) uses a key assumption that both students and studied facts are homogeneous. It assumes that students' prior knowledge in the domain can be modeled by a one-dimensional parameter.

We model the prior knowledge by the Rasch model, i.e., we have a student parameter θ_s corresponding to the global knowledge of a student s of a domain and an item parameter d_i corresponding to the difficulty of an item i. The probability that the student answers correctly is estimated using a logistic function of a difference between the global skill and the difficulty: $P(correct|\theta_s, d_i) = \sigma(\theta_s - d_i)$.

A common approach to the parameter estimation for the Rasch model is joint maximum likelihood estimation (JMLE). This is an iterative approach that is slow for large data, particularly it is not suitable for an online application, where we need to adjust estimates of parameters continuously.

In previous work [22, 25] we have shown that the parameter estimation can be done effectively using a variant of the Elo rating system [9]. The Elo rating system was originally devised for chess rating, but we can use it in student modeling by interpreting a student's answer on an item as a "match" between the student and the item. The skill and difficulty estimates are updated as follows:

$$\begin{aligned} \theta_s &:= \theta_s + K \cdot (correct - P(correct|\theta_s, d_i)) \\ d_i &:= d_i + K \cdot (P(correct|\theta_s, d_i) - correct) \end{aligned}$$

where *correct* denotes whether the question was answered correctly and K is a constant specifying sensitivity of the estimate to the last attempt. An intuitive improvement, which is used in most Elo extensions, is to use an "uncertainty function" instead of a constant K – the update should get smaller as we have more data about a student or an item. We use an uncertainty function $U(n) = \alpha/(1+\beta n)$, where n is the number of previous updates to the estimated parameters and α, β are meta-parameters.

2.2 Bayesian Model

In the basic model the uncertainty is modeled as a simple function of number of attempts. Such an approach is a simplification since some answers are more informative than others and thus the effect of answers on reduction of uncertainty should be differentiated. This can be done by using a Bayesian modeling approach. For this model we treat θ_s, d_i and *correct* as random variables. We can use Bayes' theorem for updating our beliefs about skills and difficulties:

$$P(\theta_s, d_i|correct) \propto P(correct|\theta_s, d_i) \cdot P(\theta_s, d_i)$$

We assume that the difficulty of an item is independent of a skill of a student and thus $P(\theta_s, d_i) = P(\theta_s) \cdot P(d_i)$. The updated beliefs can be expressed as marginals of the conditional distribution, for example:

$$P(\theta_s|correct) \propto P(\theta_s) \cdot \int_{-\infty}^{\infty} P(correct|\theta_s, d_i = y) \cdot P(d_i = y) \mathrm{d}y$$

In the context of rating systems for games, the basic Elo system has been extended in this direction, particularly in the Glicko system [11]. It models prior skill by a normal distribution and uses numerical approximation to represent the posterior by a normal distribution and to perform the update of the mean and standard deviation of the skill distribution using a closed form expressions. Another Bayesian extension is TrueSkill [12], which further extends the system to allow team competitions.

This approach is, however, difficult to modify for new situations, e.g., in our case we want to use the shifted logistic function (for modeling answers to multiple-choice questions), which significantly complicates derivation of equations for numerical approximation. Therefore, we use a more flexible particle based method to represent the skill distribution. The skill is represented by a skill vector $\boldsymbol{\theta_s}$, which gives the values of skill particles, and probability vector $\boldsymbol{p_s}$, which gives the probabilities of the skill particles (sums to 1). The item difficulty is represented analogically by a difficulty vector $\boldsymbol{d_i}$ and a probability vector $\boldsymbol{p_i}$. In the following text the notation $\boldsymbol{p_{s_k}}$ stands for the k-th element of the vector $\boldsymbol{p_s}$.

The skill and difficulty vectors are initialized to contain values that are spread evenly in a specific interval around zero. The probability vectors are initialized to proportionally reflect the probabilities of the particles in the selected prior distribution. During updates, only the probability vectors change, the vectors that contain the values of the particles stay fixed. Particles are updated as follows:

$$\begin{aligned} \boldsymbol{p_{s_k}} &:= \boldsymbol{p_{s_k}} \cdot \sum_{l=1}^{n} P(correct|\theta_s = \boldsymbol{\theta_{s_k}}, d_i = \boldsymbol{d_{i_l}}) \cdot \boldsymbol{p_{i_l}} \\ \boldsymbol{p_{i_l}} &:= \boldsymbol{p_{i_l}} \cdot \sum_{k=1}^{n} P(correct|\theta_s = \boldsymbol{\theta_{s_k}}, d_i = \boldsymbol{d_{i_l}}) \cdot \boldsymbol{p_{s_k}} \end{aligned}$$

After the update, we must normalize the probability vectors so that they sum to one. A reasonable simplification that avoids summing over the particle values is:

$$\begin{aligned} \boldsymbol{p_{s}}_k &:= \boldsymbol{p_{s}}_k \cdot P(correct|\theta_s = \boldsymbol{\theta_{s}}_k, d_i = E[\boldsymbol{d_i}]) \\ \boldsymbol{p_{i}}_l &:= \boldsymbol{p_{i}}_l \cdot P(correct|\theta_s = E[\boldsymbol{\theta_s}], d_i = \boldsymbol{d_{i}}_l) \end{aligned}$$

where $E[\boldsymbol{d_i}]$ ($E[\boldsymbol{\theta_s}]$) is the expected difficulty (skill) particle value (i.e. $E[\boldsymbol{d_i}] = \boldsymbol{d_i}^T \cdot \boldsymbol{p_i}$). By setting the number of particles we can trade off between precision on one hand and speed and memory requirements on the other hand.

Using the described particle model in a real-world application would require storing the probabilities for all the particles in a database. If we assume that our beliefs stay normal-like even after many observations then we can approximate each of the posteriors by a normal distribution. This approach is called assumed-density filtering [17]. Consequently, each posterior can be represented by just two numbers, the mean and the standard deviation. In this simplified model, each update requires the generation of new particles. We generate the particles in the interval $(\mu - 6\sigma, \mu + 6\sigma)$. Otherwise, the update stays the same as before. After the update is performed, the mean and the standard deviation are estimated in a standard way: $\mu_{\theta_s} := \boldsymbol{\theta_s}^T \cdot \boldsymbol{p_s}, \sigma_{\theta_s} := \|\boldsymbol{\theta_s} - \mu_{\theta_s}\|_2$.

The model can be extended to include multiplicative factors for items (q_i) and students (r_s), similarly to the Q-matrix method [1] or collaborative filtering [15]. Let k be the number of factors, then x passed in to the likelihood function $\sigma(x)$ has the form: $x = \theta_s - d_i + \sum_{j=1}^{k} q_{i,j} \cdot r_{s,j}$. The updates are similar, we only need to track more variables.

2.3 Hierarchical Model

In the next model, which we call 'hierarchical', we try to capture the domain in more detail by relaxing the assumption of homogeneity. Items are divided into disjoint sets – usually called 'concepts' or 'knowledge components' (e.g., states into continents). In addition to the global skill θ_s the model now uses also the concept skill θ_{sc}. We use an extension of the Elo system to estimate the model parameters. Predictions are done in the same way as in the basic Elo system, we just correct the global skill by the concept skill: $P(correct|\theta_s, \theta_{sc}, d_i) = \sigma((\theta_s + \theta_{sc}) - d_i)$. The update of parameters is also analogical (U is the uncertainty function and γ is a meta-parameter specifying sensitivity of the model to concepts):

$$\begin{aligned} \theta_s &:= \theta_s + U(n_s) \cdot (correct - P(correct|\theta_s, \theta_{sc}, d_i)) \\ \theta_{sc} &:= \theta_{sc} + \gamma \cdot U(n_{sc}) \cdot (correct - P(correct|\theta_s, \theta_{sc}, d_i)) \\ d_i &:= d_i + U(n_i) \cdot (P(correct|\theta_s, \theta_{sc}, d_i) - correct) \end{aligned}$$

This proposed model is related to several student modeling approaches. It can be viewed as a simplified Bayesian network model [3, 13, 16]. In a proper Bayesian network model we would model skills by a probability distribution and update the estimates using Bayes rule; equations in our model correspond to a simplification of this computation using only point skill estimates. Bayesian network model can also model more complex relationships (e.g., prerequisites), which are not necessary for our case (fact learning). Other related modeling approaches are the Q-matrix method [1], which focuses on modeling mapping between skills and items (mainly using $N : M$ relations), and models based on knowledge space theory [8]. Both these approaches are more complex than the proposed model. Our aim here is to evaluate whether even a simple concept based model is sensible for modeling factual knowledge.

The advantage of the hierarchical model is that user skill is represented in more detail and the model is thus less sensitive to the assumption of homogeneity among students. However, to use the hierarchical model, we need to determine concepts (mapping of items into groups). This can be done in several ways. Concepts may be specified manually by a domain expert. In the case of geography learning application some groupings are natural (continents, cities). In other cases the construction of concepts is more difficult, e.g., in the case of foreign language vocabulary it is not clear how to determine coherent groups of words. It is also possible to create concepts automatically or to refine expert provided concepts with the use of machine learning techniques [5, 19].

To determine concepts automatically it is possible use classical clustering methods. For our experiments we used spectral clustering method [27] with similarity of items i, j defined as a Spearman's correlation coefficient c_{ij} of correctness of answers (represented as 0 or 1) of shared students s (those who answered both items). To take into account the use of multiple-choice questions we decrease the binary representation of a response r by guess factor to $r - 1/k$ (k is the number of options). Disadvantages of the automatic concept construction are unknown number of concept, which is a next parameter to fit, and the fact that found concepts are difficult to interpret.

It is also possible to combine the manual and the automatic construction of concepts [19]. With this approach the manually constructed concepts are used as item labels. Items with these labels are used as a training set of a supervised learning method (we used logistic regression with regularization). For the item i, the vector of correlation with all items c_{ij} is used as vector of features. Errors of the used classification method are interpreted as "corrected" labels; see [19, 20] for more details.

2.4 Networked Model

The hierarchical model enforces hard division of items into groups. With the next model we bypass this division by modeling directly relations among individual items, i.e., we treat items as a network (and hence the name 'networked model'). For each item we have a local skill θ_{si}. For each pair of items we compute the degree to which they are correlated c_{ij}. This is done from training data or – in the real system – once a certain number of answers is collected. After the answer to the item i all skill estimates for all other items j are updated based on c_{ij}. The model still uses the global skill θ_s and makes the final prediction based on the weighted combination of global and local skill: $P(correct|\theta_s, \theta_{si}) = \sigma(w_1\theta_s + w_2\theta_{si} - d_i)$. Parameters are updated as follows:

$$\begin{aligned} \theta_s &:= \theta_s + U(n_s) \cdot (correct - P(correct|\theta_s, \theta_{si})) \\ \theta_{sj} &:= \theta_{sj} + c_{ij} \cdot U(n_s) \cdot (correct - P(correct|\theta_s, \theta_{si})) \\ &\quad \text{for all items } j \\ d_i &:= d_i + U(n_i) \cdot (P(correct|\theta_s, \theta_{si}) - correct) \end{aligned}$$

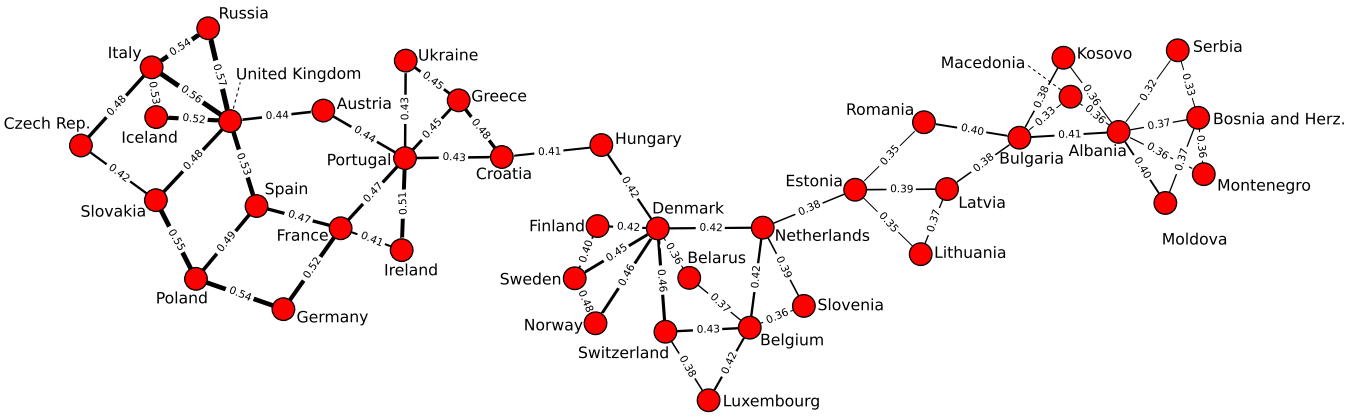


Figure 1: Illustration of the networked model on European countries. Only the most important edges for each country are shown.

This model is closely related to multivariate Elo which was previously proposed in the context of adaptive psychometric experiments [7].

For illustration of the model, Figure 1 shows selection of the most important correlations for European countries. Note that this automatically generated figure contains some natural clusters as Balkan countries (right), Scandinavian countries (middle), and well-known[1] countries (left).

3. EVALUATION

We provide evaluation of the above described models over data from an adaptive application for learning facts.

3.1 The Used System and Data

For the analysis we use data from an online adaptive system `slepemapy.cz` for practice of geography facts (e.g., names and location of countries, cities, mountains). The system estimates student knowledge and based on this estimate it adaptively selects questions of suitable difficulty [22]. The system uses a target success rate (e.g., 75 %) and adaptively selects questions in such a way that the students' achieved performance is close to this target [21]. The system uses open questions ("Where is France?") and multiple-choice questions ("What is the name of the highlighted country?") with 2 to 6 options. Students answer questions with the use of an interactive 'outline map'. Students can also access a visualization of their knowledge using an open learner model.

Our aim is to model prior knowledge (not learning during the use of the system), so we selected only the first answers of students to every item. The used data set contains more than 1.8 million answers of 43 thousand students. The system was originally available only in Czech, currently it is available in Czech, English, and Spanish, but students are still mostly from Czech republic ($> 85\%$) and Slovakia ($> 10\%$). The data set was split into train set (30%) and test set (70%) in a student-stratified manner. As a primary metric for model comparison and parameter fitting we use root mean square error (RMSE), since the application works with absolute values of predictions [22] (see [26] for more details on choice of a metric).

[1]By students using our system.

3.2 Model Parameters

The train set was used for finding the values of the meta-parameters of individual models. Grid search was used to search the best parameters of the uncertainty function $U(n)$. Left part of Figure 2 shows RMSE of the basic Elo model on training data for various choices of α and β. We chose $\alpha = 1$ and $\beta = 0.06$ and we used these values also for derived models which use the uncertainty function.

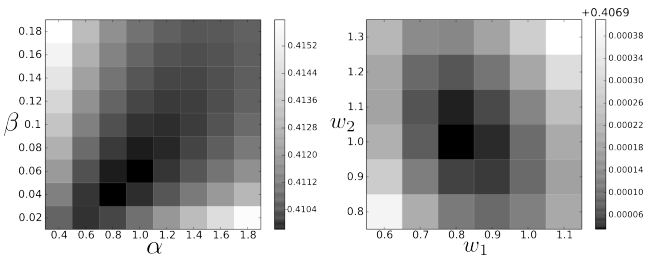


Figure 2: Grid searches for the best uncertainty function parameters α, β (left) and the best parameters w_1, w_2 of the networked model (right). As can be seen from different scales, models are more sensitive to α and β parameters.

Grid search (Figure 2 right) was used also to find the best parameters $w_1 = 0.8, w_2 = 1$ of the networked model. The train set was also used for computation of correlations. To avoid spurious high correlations of two items i, j as consequence of lack of common students we set all $c_{ij} = 0$ for those pairs i, j with less than 200 common students. Correlations computed by this method show stability with respect to selection of train set. For two different randomly selected train sets correlation values correlate well (> 0.95). As Figure 1 shows, the resulting correlations are interpretable.

For the particle-based Bayesian model we can tune the performance by setting the number of particles it uses for estimating each distribution. We found out that increasing the number of particles beyond 100 does not increase performance. For the simplified version, only 10 particles are sufficient. This is probably due to the way the algorithm uses the particles (they are discarded after each step).

Table 1: Comparison of models on the test set.

Model	RMSE	LL	AUC
Elo ($\alpha = 1, \beta = 0.06$)	0.4076	−643179	0.7479
Bayesian model	0.4080	−644362	0.7466
Bayesian model (3 skills)	0.4056	−637576	0.7533
Hierarchical model	0.4053	−636630	0.7552
Networked model	0.4053	−636407	0.7552

3.3 Accuracy of Predictions

All the reported models work online. Training of models (parameters θ_s, d_i) continues on the test set but only predictions on this set are used to evaluate models.

Table 1 shows results of model comparison with respect model performance metrics. In addition to RMSE we also report log-likelihood (LL) and area under the ROC curve (AUC); the main result are not dependent on the choice of metric. In fact, predictions for individual answers are highly correlated. For example for the basic Elo model and hierarchical model most of the predictions (95%) differ by less than 0.1.

The hierarchical model reported in Table 1 uses manually determined concepts based on both location (e.g., continent) and type of place (e.g., country). Both the hierarchical model and the networked model bring an improvement over the basic Elo model. The improvement is statistically significant (as determined by a t-test over results of repeated cross-validation), but it is rather small. Curiously, the Particle Bayes model is slightly worse than the simple Elo system, i.e., the more involved modeling of uncertainty does not improve predictions. The performance improves only when we use the multiple skill extension. We hypothesize that the improvement of the hierarchical (resp. multiple skill) extensions model be more significant for less homogeneous populations of students. Each skill could then be used to represent a different prior knowledge group.

RMSE is closely related to Brier score [26], which provides decomposition [18] to uncertainty (measures the inherent uncertainty in the observed data), reliability (measures how close the predictions are to the true probabilities) and resolution (measures how diverse the predictions are).

This decomposition can be also illustrated graphically. Figure 3 shows comparison of the basic Elo model and the hierarchical model. Both calibration lines (which are near the optimal one) reflect very good reliability. On the other hand, histograms reflect the fact that the hierarchical model gives more divergent predictions and thus has better resolution.

3.4 Using Models for Insight

In student modeling we are interested not just in predictions, but also in getting insight into characteristics of the domain or student learning. The advantage of more complex models may lie in additional parameters, which bring or improve such insight.

Figure 5 gives comparison of item difficulty for Elo model

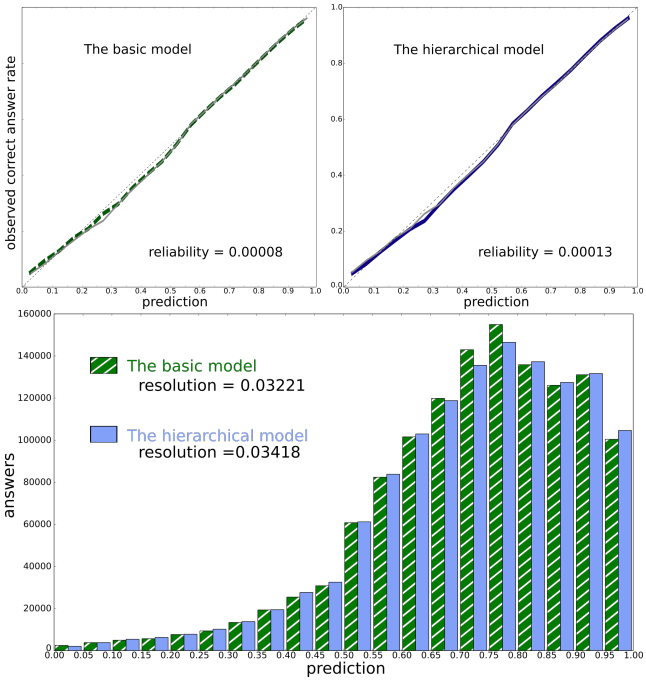


Figure 3: Illustration of the Brier score decomposition for the basic model and the hierarchical model. Top: reliability (calibration curves). Bottom: resolution (histograms of predicted values).

and Particle Bayes. As we can see, the estimated values of the difficulties are quite similar. The main difference between these models is in estimates of uncertainty. The uncertainty function used in Elo converges to zero faster and its shape is the same for all items. In Particle Bayes, the uncertainty is represented by the standard deviation of the normal distribution. This uncertainty can decrease differently for each item, depending on the amount of surprising evidence the algorithm receives, as is shown in Figure 4. The better grasp of uncertainty can be useful for visualization in an open learner model [2, 10].

Other extensions (networked, hierarchical, Bayesian with multiple skills) bring insight into the domain thanks to the analysis of relations between items, e.g., by identifying most

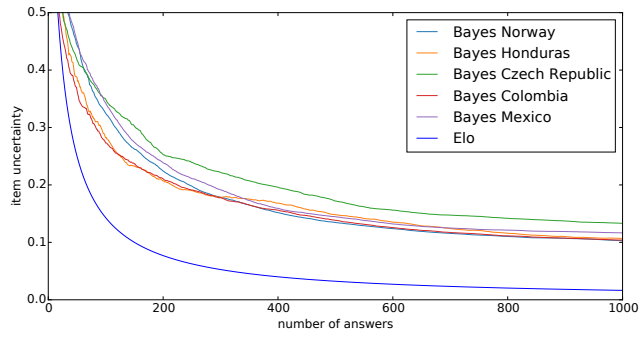


Figure 4: Evolution of uncertainties in the Bayes model and Elo.

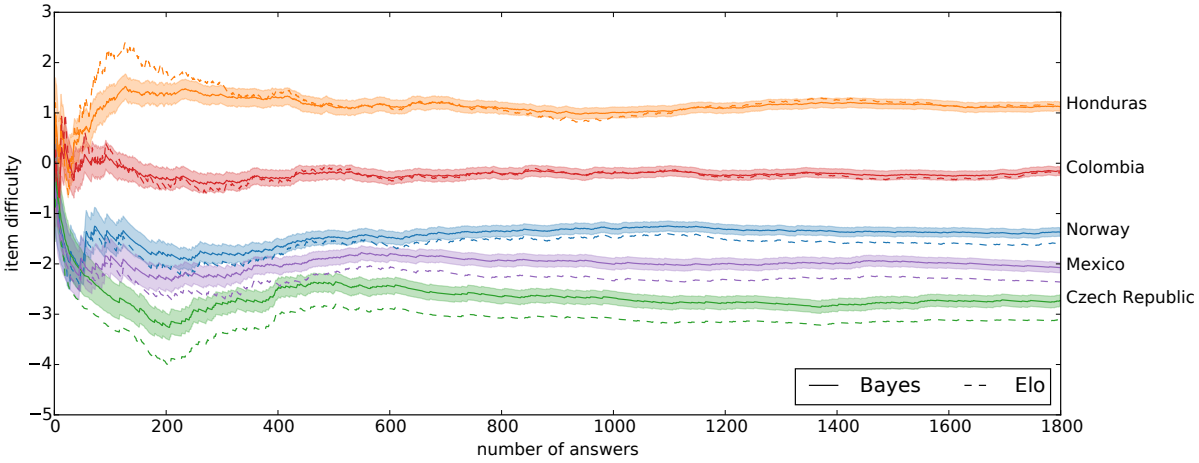


Figure 5: Difficulty of countries – the basic Elo model versus the Bayes model.

useful clusters of items. Such results can be used for improving the behavior of an adaptive educational system. For example, the system can let the user practice items from one concept and after reaching mastery move to the next one. Another possible use of concepts is for automatic construction of multiple-choice questions with good distractors (falling under the same concept).

We performed evaluation of the hierarchical model with different concepts. We used several approaches for specifying the concepts manually: based on type (e.g., countries, cities, rivers), location (e.g., Europe, Africa, Asia) and combination of the two approaches (e.g, European countries, European cities, African countries). Since we have most students' answers for European countries, we also considered a data set containing only answers on European countries. For this data set we used two sets of concepts. The first is the partition to Eastern, Western, Northwestern, Southern, Central and Southeastern Europe, the second concept set is obtained from the first one by union of Central, Western and Southern Europe (countries from these regions are mostly well-known by our Czech students) and union of Southeastern and Eastern Europe.

We compared these manually specified concepts with automatically corrected and entirely automatically constructed concepts (as described in Section 2.3; 'corrected' concepts are based on manually specified concepts and are revised based on the data). The quality of concepts was evaluated using prediction accuracy of the hierarchical model which uses these concepts. Table 2 shows the results expressed as RMSE improvement over the basic model. Note that the differences in RMSE are necessarily small, since the used models are very similar and differ only in the allocation of items to concepts. For the whole data set (1368 items) a larger number of concepts brings improvement of performance. The best results are achieved by manually specified concepts (combination of location and type of place), automatic correction does not lead to significantly different performance. For the smaller data set of European countries (39 items) a larger number of (both manual and automatically determined) concepts brings worse performance – a model with too small concepts suffers from a loss of information. In this case the best result is achieved by a correction of manually specified concepts. The analysis shows that the corrections make intuitive sense, most of them are shifts of well-known and easily recognizable countries as Russia or Iceland to block of well-known countries (union of Central, Western and Southern Europe).

Table 2: Comparison of manual, automatically corrected manual, and automatic concepts. Quality of concepts is expressed as RMSE improvement of the hierarchical model with these concepts over the basic model.

All items	number of concepts	RMSE improvement
manual – type	14	0.00132
corrected – type	14	0.00132
manual – location	22	0.00179
corrected – location	22	0.00167
manual – combination	**56**	**0.00235**
corrected – combination	56	0.00234
automatic	5	−0.00025
automatic	20	0.00039
automatic	50	0.00057
Europe		
manual	3	0.00003
corrected	**3**	**0.00011**
manual	6	−0.00015
corrected	6	0.00003
automatic	2	0.00007
automatic	3	0.00004
automatic	5	−0.00019

Models with multiple skills bring some additional information not just about the domain, but also about students. Correlation of concept skills with the global skill range from -0.1 to 0.5; the most correlated concepts are the ones with large number of answers like European countries (0.48) or

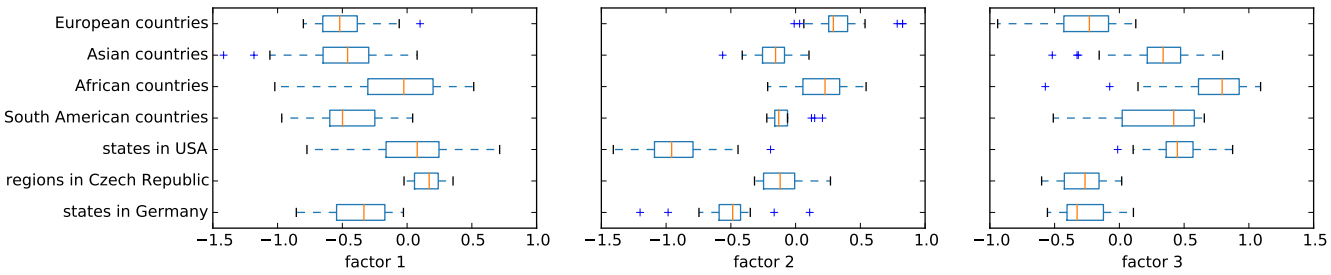


Figure 6: Boxplots of the item factor values from the Bayesian model (3 factors) grouped by some manually created concepts.

Asian countries (0.4), since answers on items in these concepts have also large influence on the global skill. Correlation between two clusters skills typically range from -0.1 to 0.1. These low correlation values suggest that concept skills hold interesting additional information about student knowledge.

Another view of relations between items is provided by the Bayesian model with multiplicative factors – this model does not provide division of items into disjoint sets, but rather determines for each item a strength of its relation to each factor (based on the data). Figure 6 illustrates how the learned factors relate to some manually specified concepts. Note that the results in Table 1 suggest that most of the improvement in predictive accuracy can be achieved by just these three automatically constructed factors. We can see that *factor 3* discriminates well between countries in Europe and Africa (Figure 7 provides a more detailed visualization). In the case of geography the division of items to concepts can be done in rather natural way and thus the potential application of such automatically determined division is limited and serves mainly as a verification of the method. For other domains (e.g., vocabulary learning) such natural division may not exist and this kind of model output can be very useful.

Also, note that *Factor 2* differentiates between states in USA and countries on other continents and *Factors 1* and *2* have different values for regions in Czech republic and states in Germany. This evidence supports an assumption that the model may be able to recognize students with varied background.

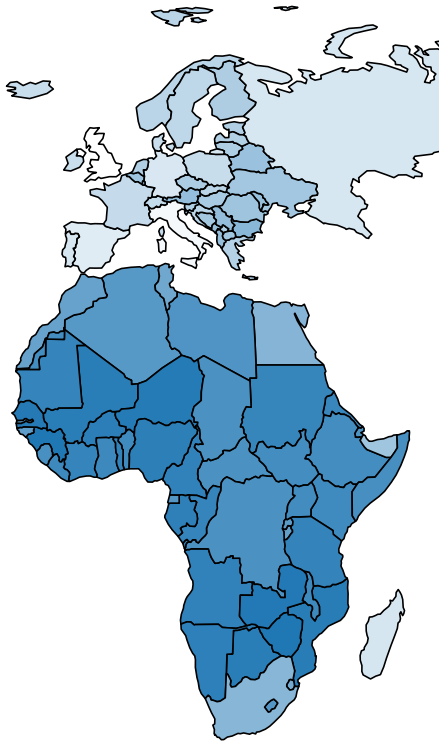


Figure 7: Visualization of the values of the third factor in the Bayesian model with multiple skills.

4. DISCUSSION

We have described and compared several student models of prior knowledge. The models were evaluated over extensive data from application for learning geography. The described models should be directly applicable to other online systems for learning facts, e.g., in areas like biology, human anatomy, or foreign language vocabulary. For application in domains which require deeper understanding (e.g., mathematics, physics) it may be necessary to develop extensions of described models (e.g., to capture prerequisite relations among concepts).

The results show that if we are concerned only with the accuracy of predictions, the basic Elo model is a reasonable choice. More complex models do improve predictions in statistically significant way, but the improvement is relatively small and evenly spread (i.e., individual predictions by different models are very similar).

The improvement in predictions by the hierarchical or networked models may be more pronounced in less homogeneous domains or with less homogeneous populations. Nevertheless, if the main aim of a student model is prediction of future answers (e.g., applied for selection of question), then the basic Elo model seems to be sufficient. Its performance is good and it is very simple to apply. Thus, we believe that it should be used more often both in implementations of educational software and in evaluations of student models.

The more complex models may still be useful, since improved accuracy is not the only purpose of student models. Described models have interpretable parameters – assignment of items to concepts and better quantification of uncertainty

of estimates of knowledge and difficulty. These parameters may be useful by themselves. We can use them to guide the adaptive behavior of educational systems, e.g., the choice of questions can be done in such a way that it respects the determined concepts or at the beginning of the session we can prefer items with low uncertainty (to have high confidence in choosing items with appropriate difficulty). The uncertainty parameter is useful for visualization of student knowledge in open learner models [2, 10]. Automatically determined concepts may also provide useful feedback to system developers, as they suggest potential improvements in user interface, and also to teachers for whom they offer insight into student's (mis)understanding of target domain. Given the small differences in predictive accuracy, future research into extensions of basic models should probably focus on these potential applications.

5. REFERENCES

[1] Tiffany Barnes. The q-matrix method: Mining student response data for knowledge. In *Educational Data Mining*, 2005.

[2] Susan Bull. Supporting learning with open learner models. In *Information and Communication Technologies in Education*, 2004.

[3] Cristina Conati, Abigail Gertner, and Kurt Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4):371–417, 2002.

[4] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[5] Michel C Desmarais, Behzad Beheshti, and Rhouma Naceur. Item to skills mapping: deriving a conjunctive q-matrix from data. In *Intelligent Tutoring Systems*, pages 454–463. Springer, 2012.

[6] Michel C Desmarais and Ryan SJ d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[7] Philipp Doebler, Mohsen Alavash, and Carsten Giessing. Adaptive experiments with a multivariate elo-type algorithm. *Behavior Research Methods*, pages 1–11, 2014.

[8] Jean-Paul Doignon and Jean-Claude Falmagne. *Knowledge spaces*. Springer, 1999.

[9] Arpad E Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[10] Carrie Demmans Epp, Susan Bull, and Matthew D Johnson. Visualising uncertainty for open learner model users. 2014. to appear.

[11] Mark E Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

[12] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.

[13] Tanja Käser, Severin Klingler, Alexander Gerhard Schwing, and Markus Gross. Beyond knowledge tracing: Modeling skill topologies with bayesian networks. In *Intelligent Tutoring Systems*, pages 188–198. Springer, 2014.

[14] S Klinkenberg, M Straatemeier, and HLJ Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[15] Yehuda Koren and Robert Bell. Advances in collaborative filtering. *Recommender Systems Handbook*, pages 145–186, 2011.

[16] Eva Millán, Tomasz Loboda, and Jose Luis Pérez-de-la Cruz. Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683, 2010.

[17] Thomas P Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[18] Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.

[19] Juraj Nižnan, Radek Pelánek, and Jiří Řihák. Using problem solving times and expert opinion to detect skills. In *Educational Data Mining (EDM)*, pages 434–434, 2014.

[20] Juraj Nižnan, Radek Pelánek, and Jiří Řihák. Mapping problems to skills combining expert opinion and student data. In *Mathematical and Engineering Methods in Computer Science*, pages 113–124. Springer, 2014.

[21] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Artificial Intelligence in Education*, 2015.

[22] Jan Papoušek, Radek Pelánek, and Vít Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining (EDM)*, pages 6–13, 2014.

[23] Zachary A Pardos and Neil T Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[24] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[25] Radek Pelánek. Time decay functions and elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.

[26] Radek Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 2015. To appear.

[27] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Discovering Prerequisite Structure of Skills through Probabilistic Association Rules Mining

Yang Chen, Pierre-Henri Wuillemin, Jean-Marc Labat
Sorbonne Universites, UPMC, Univ. Paris 6, UMR 7606, LIP6, Paris, France
CNRS, UMR 7606, CNRS, Paris, France
4 Place Jussieu, 75005 Paris, France
{yang.chen, pierre-henri.wuillemin, jean-marc.labat}@lip6.fr

ABSTRACT

Estimating the prerequisite structure of skills is a crucial issue in domain modeling. Students usually learn skills in sequence since the preliminary skills need to be learned prior to the complex skills. The prerequisite relations between skills underlie the design of learning sequence and adaptation strategies for tutoring systems. The prerequisite structures of skills are usually studied by human experts, but they are seldom tested empirically. Due to plenty of educational data available, in this paper, we intend to discover the prerequisite structure of skills from student performance data. However, it is a challenging task since skills are latent variables. Uncertainty exists in inferring student knowledge of skills from performance data. Probabilistic Association Rules Mining proposed by Sun et al. (2010) is a novel technique to discover association rules from uncertain data. In this paper, we preprocess student performance data by an evidence model. Then the probabilistic knowledge states of students estimated by the evidence model are used by the probabilistic association rules mining to discover the prerequisite structure of skills. We adapt our method to the testing data and the log data with different evidence models. One simulated data set and two real data sets are used to validate our method. The discovered prerequisite structures can be provided to assist human experts in domain modeling or to validate the prerequisite structures of skills from human expertise.

Keywords

Probabilistic association rules mining, Skill structure, Prerequisite, DINA, BKT

1. INTRODUCTION

In most Intelligent Tutoring Systems (ITSs) and other educational environments, learning sequence is an important issue investigated by many educators and researchers. It is widely believed that students should be capable of solving the easier problems before the difficult ones are presented to them, and likewise, some preliminary skills should be learned prior to the learning of the complex skills. The prerequisite relations between problems and between skills underlie the adaptation strategies for tutoring and assessments. Furthermore, improving the accuracy of a student model with the prerequisite structure of skills has been exemplified by [1, 2]. The prerequisite structures of problems and skills are in accordance with the Knowledge Space Theory [3] and Competence-based Knowledge Space Theory [4]. A student's knowledge state should comply with the prerequisite structure of skills. If a skill is mastered by a student, all the prerequisites of the skill should also be mastered by the student. If any prerequisite of a skill is not mastered by a student, it seems difficult for the student to learn the skill. Therefore, according to the knowledge states of students, we can uncover the prerequisite structure of skills. Most prerequisite structures of skills reported in the student modeling literature are studied by domain or cognition experts. It is a tough and time-consuming task since it is quite likely that the prerequisite structures from different experts on the same set of skills are difficult to come to an agreement. Moreover, the prerequisite structures from domain experts are seldom tested empirically. Nowadays, some prevalent data mining and machine learning techniques have been applied in cognition models, benefiting from large educational data available through online educational systems. Deriving the prerequisite structures of observable variables (e.g. problems) from data has been investigated by some researchers. However, discovering prerequisite structures of skills is still challenging since a student's knowledge of a skill is a latent variable. Uncertainty exists in inferring student knowledge of skills from performance data. This paper aims to discover the prerequisite structures of skills from student performance data.

2. RELATED WORK

With the emerging educational data mining techniques, many works have investigated the discovery of the prerequisite structures within domain models from data. The Partial Order Knowledge Structures (POKS) learning algorithm is proposed by Desmarais and his colleagues [5] to learn the item to item knowledge structures (i.e. the prerequisite structure of problems) which are solely composed of the observable nodes, like answers to test questions. The results from the experiments over their three data sets show that the POKS algorithm outperforms the classic BN structure learning algorithms [6] on the predictive ability and the computational efficiency. Pavlik Jr. et al. [7] used the POKS algorithm to analyze the relationships between the observable item-type skills, and the results were used for the hierarchical agglomerative clustering to improve the skill model. Vuong et al. [8] proposed a method to determine the dependency relationships between units in a curriculum with the student performance data that are observed at the unit level (i.e. graduating from a unit or not). They used the statistic binominal test to look for a significant difference between the performance of students who used the potential prerequisite unit and the performance of students who did not. If a significant difference is found, the prerequisite relation is deemed to exist. All these methods above are proposed

to discover prerequisite structures of the observable variables. Tseng et al. [9] proposed to use the frequent association rules mining to discover concept maps. They constructed concept maps by mining frequent association rules on the data of the fuzzy grades from students' testing. They used a deterministic method to transfer frequent association rules on questions to the prerequisite relations between concepts, without considering the uncertainty in the process of transferring students' performance to their knowledge. Deriving the prerequisite structure of skills from noisy observations of student knowledge is considered in the approach of Brunskill [10]. In this approach, the log likelihood is computed for the precondition model and the flat model (skills are independent) on each skill pair to estimate which model better fits the observed student data. Scheines et al. [11] extended causal discovery algorithms to discover the prerequisite structure of skills by performing statistical tests on latent variables. In this paper, we propose to apply a data mining technique, namely the probabilistic association rules mining, to discover prerequisite structures of skills from student performance data.

3. METHOD

Association rules mining [12] is a well-known data mining technique for discovering the interesting association rules in a database. Let $I=\{i_1,i_2,\cdots,i_m\}$ be a set of attributes (called items) and $D=\{r_1,r_2,\cdots,r_n\}$ be a set of records (or transactions), i.e. a database. Each record contains the values for all the attributes in I. A pattern (called itemset) contains the values for some of the attributes in I. The support count of pattern X is the number of records in D that contain X, denoted by $\sigma(X)$. An association rule is an implication of the form $X \Rightarrow Y$, where X and Y are related to the disjoint sets of attributes. Two measures are commonly used to discover the strong or interesting association rules: the support of rule $X \Rightarrow Y$ denoted by $Sup(X \Rightarrow Y)$, which is the percentage of records in D that contain $X \cup Y$, i.e. $P(X \cup Y)$; the confidence denoted by $Conf(X \Rightarrow Y)$, which is the percentage of records in D containing X that also contains Y, i.e. $P(Y|X)$. The rule $X \Rightarrow Y$ is considered strong or interesting if it satisfies the following condition:

$$(Sup(X \Rightarrow Y) \geq minsup) \wedge (Conf(X \Rightarrow Y) \geq minconf) \quad (1)$$

where *minsup* and *minconf* denote the minimum support threshold and the minimum confidence threshold. The support threshold is used to discover frequent patterns in a database, and the confidence threshold is used to discover the association rules within the frequent patterns. The support condition makes sure the coverage of the rule, that is, there are adequate records in the database to which the rule applies. The confidence condition guarantees the accuracy of applying the rule. The rules which do not satisfy the support threshold or the confidence threshold are discarded in consideration of the reliability. Consequently, the strong association rules could be selected by the two thresholds.

To discover the skill structure, a database of students' knowledge states is required. The knowledge state of a student is a record in the database and the mastery of a skill is a binary attribute with the values mastered (1) and non-mastered (0). If skill Si is a prerequisite of skill Sj, it is most likely that Si is mastered given that Sj is mastered, and that skill Sj is not mastered given that Si is not mastered. Thus this prerequisite relation corresponds with the two association rules: $Sj=1 \Rightarrow Si=1$ and $Si=0 \Rightarrow Sj=0$. If both the association rules exist in a database, Si is deemed a prerequisite of Sj. To examine if both the association rules exist in a database, according to condition (1), the following conditions could be used:

$$(Sup(Sj=1 \Rightarrow Si=1) \geq minsup) \wedge (Conf(Sj=1 \Rightarrow Si=1) \geq minconf) \quad (2)$$

$$(Sup(Si=0 \Rightarrow Sj=0) \geq minsup) \wedge (Conf(Si=0 \Rightarrow Sj=0) \geq minconf) \quad (3)$$

When condition (2) is satisfied, the association rule $Sj=1 \Rightarrow Si=1$ is deemed to exist in the database, and when the condition (3) is satisfied, the association rule $Si=0 \Rightarrow Sj=0$ is deemed to exist in the database. Theoretically, if skill Si is a prerequisite of Sj, all the records in the database should comply with the two association rules. To be exact, the knowledge state $\{Si=0, Sj=1\}$ should be impossible, thereby $\sigma(Si=0, Sj=1)$ should be 0. According to the equations (4) and (5), the confidences of the rules in the equations should be 1.0. Since noise always exists in real situations, when the confidence of an association rule is greater than a threshold, the rule is considered to exist if the support condition is also satisfied. We cannot conclude that the prerequisite relation exists if one rule exists but the other not. For instance, the high confidence of the rule $Sj=1 \Rightarrow Si=1$ might be caused by the high proportion $P(Si=1)$ in the data.

$$Conf(Sj=1 \Rightarrow Si=1) = P(Si=1|Sj=1) = \frac{\sigma(Si=1,Sj=1)}{\sigma(Si=1,Sj=1)+\sigma(Si=0,Sj=1)} \rightarrow 1 \quad (4)$$

$$Conf(Si=0 \Rightarrow Sj=0) = P(Sj=0|Si=0) = \frac{\sigma(Si=0,Sj=0)}{\sigma(Si=0,Sj=0)+\sigma(Si=0,Sj=1)} \rightarrow 1 \quad (5)$$

The discovery of the association rules within a database depends on the support and confidence thresholds. When the support threshold is given a relatively low value, more skill pairs will be considered as frequent patterns. When the confidence threshold is given a relatively low value, the weak association rules within frequent patterns will be deemed to exist. As a result, the weak prerequisite relations will be discovered. It is reasonable that the confidence threshold should be higher than 0.5. The selection of the two thresholds requires human expertise. Given the data about the knowledge states of a sample of students, the frequent association rules mining can be used to discover the prerequisite relations between skills.

However, a student's knowledge state cannot be directly obtained since student knowledge of a skill is a latent variable. In common scenarios, we collect the performance data of students in assessments or tutoring systems and estimate their knowledge states according to the observed data. The evidence models that transfer the performance data of students to their knowledge states in consideration of the noise have been investigated for several decades. The psychometric models DINA (Deterministic Input Noisy AND) and NIDA (Noisy Input Deterministic AND) [13] have been used to infer the knowledge states of students from their response data on the multi-skill test items. The well-known Bayesian Knowledge Tracing (BKT) model [14] is a Hidden Markov model that has been used to update students' knowledge states according to the log files of their learning in a tutoring system. A Q-matrix which represents the items to skills mapping is required in these models. The Q-matrix is usually created by domain experts, but recently some researchers [15, 16, 17] investigated to extract an optimal Q-matrix from data. Our method

assumes that an accurate Q-matrix is known, like the method in [11]. Since the noise (e.g. slipping and guessing) is considered in the evidence models, the likelihood that a skill is mastered by a student can be estimated. The estimated knowledge state of a student is probabilistic, which incorporates the probability of each skill mastered by the student. Table 1 shows an example of the database consisting of probabilistic knowledge states. For example, the probabilities that skills *S1*, *S2* and *S3* are mastered by student "st1" are 0.9, 0.8 and 0.9 respectively.

We discover the prerequisite relations between skills from the probabilistic knowledge states of students that are estimated by an evidence model. The frequent association rules mining can no longer be used to discover the prerequisite relations between skills from a probabilistic database. Because any attribute value in a probabilistic database is associated with a probability. A probabilistic database can be interpreted as a set of deterministic instances (named possible worlds) [18], each of which is associated with a probability. We assume that the noise (e.g. slipping, guessing) causing the uncertainty for different skills is mutually independent. In addition, we assume that the knowledge states of different students are observed independently. Under these assumptions, the probability of a possible world in our database is the product of the probabilities of the attribute values over all the records in the possible world [18, 19, 20]. For example, a possible world for the database in Table 1 is that both the knowledge states of the students "st1" and "st2" are {*S1*=1, *S2*=0, *S3*=1}, whose probability is about 0.0233 (i.e. 0.9×0.2×0.9×0.2×0.9×0.8). The support count of a pattern in a probabilistic database should be computed with all the possible worlds. Thus the support count is no longer a deterministic number but a discrete random variable. Figure 1 depicts the probability mass function (*pmf*) of the support count of pattern {*S1*=1, *S2*=1} in the database of Table 1. For instance, the probability of $\sigma(S1=1, S2=1)=1$ is about 0.7112, which is the sum of the probabilities of all the possible worlds in which only one record contains the pattern {*S1*=1, *S2*=1}. Since there are an exponential number of possible worlds in a probabilistic database (e.g. 2^6 possible worlds in the database of Table 1), computing the support count of a pattern is expensive. The Dynamic-Programming algorithm proposed by Sun et al. [20] is used to efficiently compute the support count *pmf* of a pattern.

Table 1. A database of probabilistic knowledge states

Student ID	Probabilistic Knowledge State
st1	{S1: 0.9, S2: 0.8, S3: 0.9}
st2	{S1: 0.2, S2: 0.1, S3: 0.8}

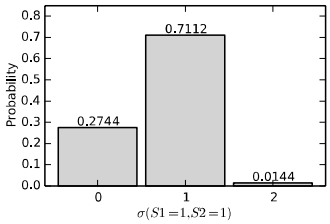


Figure 1. The support count *pmf* of the pattern {*S1*=1, *S2*=1} in the database of Table 1

To discover the prerequisite relations between skills from the probabilistic knowledge states of students, the probabilistic association rules mining technique [20] is used in this paper, which is an extension of the frequent association rules mining to discover association rules from uncertain data. Since the support count of a pattern in a probabilistic database is a random variable, the conditions (2) and (3) are satisfied with a probability. Hence the association rules derived from a probabilistic database are also probabilistic. We use the formula proposed by [20] to compute the probability of an association rule satisfying the two thresholds. It can be also interpreted as the probability of a rule existing in a probabilistic database. For instance, the probability of the association rule $Sj=1 \Rightarrow Si=1$ existing in a probabilistic database is the probability that the condition (2) is satisfied in the database:

$$\begin{aligned} & P(Sj=1 \Rightarrow Si=1) \\ & = P((Sup(Sj=1 \Rightarrow Si=1) \geq minsup) \wedge (Conf(Sj=1 \Rightarrow Si=1) \geq minconf)) \\ & = \sum_{n=minsup \times N}^{N} f_{Si=1,Sj=1}[n] \sum_{m=0}^{\frac{(1-minconf)n}{minconf}} f_{Si=0,Sj=1}[m] \end{aligned} \quad (6)$$

where N is the number of records in the database and f_X denotes the support count *pmf* of pattern X, and $f_X[k]=P(\sigma(X)=k)$.

The probability of the rule related to condition (3) is computed similarly. According to formula (6), the probability of an association rule changes with the support and confidence thresholds. Given the two thresholds, the probability of an association rule existing in a probabilistic database can be computed. And if the probability is very close to 1.0, the association rule is considered to exist in the database. If both the association rules related to a prerequisite relation are considered to exist, the prerequisite relation is considered to exist. We can use another threshold, the minimum probability threshold denoted by *minprob*, to select the most possible association rules. Thus, if both $P(Sj=1 \Rightarrow Si=1) \geq minprob$ and $P(Si=0 \Rightarrow Sj=0) \geq minprob$ are satisfied, *Si* is deemed a prerequisite of *Sj*. When a pair of skills are estimated to be the prerequisite of each other, the relation between them are symmetric. It means that the two skills are mastered or not mastered simultaneously. The skill models might be improved by merging the two skills with the symmetric relation between them.

4. EVALUATION

We use one simulated data set and two real data sets to validate our method. The prerequisite structure derived from the simulated data is compared with the presupposed structure that is used to generate the data, while the prerequisite structure derived from the real data is compared with the structure investigated by another research on the same dataset or the structure from human expertise. Moreover, we adapt our method to the testing data and the log data. Different evidence models are used to preprocess the two types of data to get the probabilistic knowledge states of students. The DINA model is used for the testing data, whereas the BKT model is used for the log data.

4.1 Simulated Testing Data

Data set. We use the data simulation tool available via the R package CDM [21] to generate the dichotomous response data according to a cognitive diagnosis model (the DINA model used here). The prerequisite structure of the four skills is presupposed as Figure 3(a). According to this structure, the knowledge space decreases to be composed of six knowledge states, that is ∅, {*S1*}, {*S1*, *S2*}, {*S1*, *S3*}, {*S1*, *S2*, *S3*}, {*S1*, *S2*, *S3*, *S4*}. The reduced knowledge space implies the prerequisite structure of the skills. The knowledge states of 1200 students are randomly generated from the reduced knowledge space restricting every knowledge state type in the same proportion (i.e. 200 students per type). The

simulated knowledge states are used as the input of the data simulation tool. There are 10 simulated testing questions, each of which requires one or two of the skills for the correct response. The slip and guess parameters for each question are restricted to be randomly selected in the range of 0.05 and 0.3. According to the DINA model with these specified parameters, the data simulation tool generates the response data. Using the simulated response data as the input of a flat DINA model, the slip and guess parameters of each question in the model are estimated and the probability of each student's knowledge on each skill is computed. The tool for the parameter estimation of DINA model is also available through the R package CDM [21], which is performed by the Expectation Maximization algorithm to maximize the marginal likelihood of data.

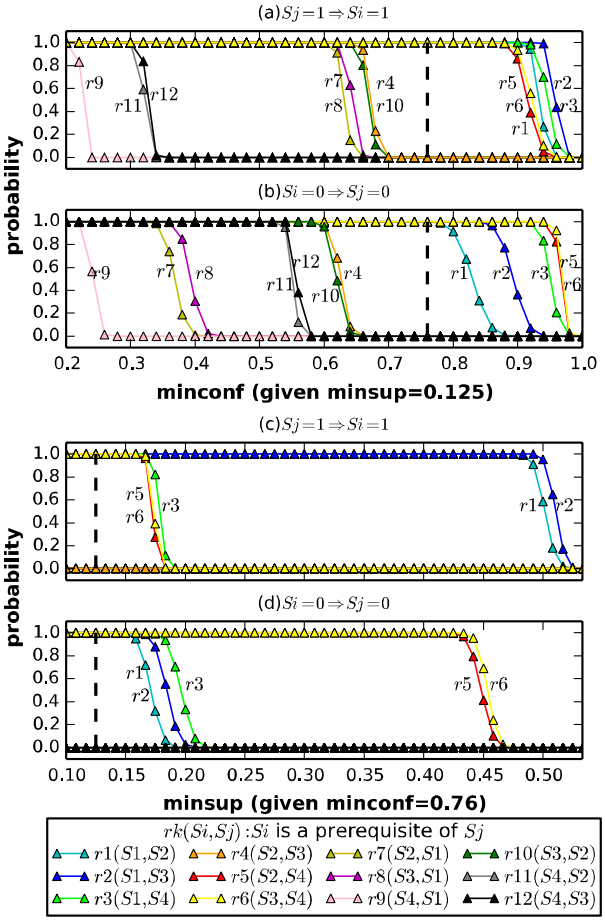


Figure 2. The probabilities of the association rules in the simulated data given different confidence or support thresholds

Result. The estimated probabilistic knowledge states of the simulated students are used as the input data to discover the prerequisite relations between skills. For each skill pair, there are two prerequisite relation candidates. For each prerequisite relation candidate, we examine if the two corresponding association rules Sj=1⇒Si=1and Si=0⇒Sj=0 exist in the database. The probability of an association rule existing in the database is computed according to formula (6), which is jointly affected by the selected support and confidence thresholds. For the sake of clarity, we look into the effect of one threshold leaving the other one unchanged. The joint effect of the two thresholds will be discussed in section 4.4. Giving a small constant to one threshold that all the association rules satisfy (perhaps several trials are needed or simply assign 0.0), we can observe how the probabilities of the association rules change with different values of the other threshold.

Figure 2 (a) and (b) describe how the probabilities of the corresponding association rules in the simulated data change with different confidence thresholds, where the support threshold is given as a constant (0.125 here). When the probability of a rule is close to 1.0, the rule is deemed to satisfy the thresholds. All the association rules satisfy the support threshold since their probabilities are almost 1.0 at first. The rules in the two figures corresponding to the same prerequisite relation candidate are depicted in the same color. In the figures, when the confidence threshold varies from 0.2 to 1.0, the probabilities of the different rules decrease from 1.0 to 0.0 in different intervals of threshold value. When we choose different threshold values, different sets of rules will be discovered. In each figure, there are five rules that can satisfy the significantly higher threshold. Given minconf=0.78, the probabilities of these rules are almost 1.0 whereas others are almost 0.0. These rules are very likely to exist. Moreover, the discovered rules in the two figures correspond to the same set of prerequisite relation candidates. Accordingly, these prerequisite relations are very likely to exist. To make sure the coverage of the association rules satisfying the high confidence threshold, it is necessary to know the support distributions of these rules. Figure 2 (c) and (d) illustrate how the probabilities of the corresponding association rules change with different support thresholds. The confidence threshold is given as a constant 0.76, and five association rules in each figure satisfy this threshold. Only on these rules, the effect of different support thresholds can be observed. In each figure, the rules gather in two intervals of threshold value. For example, in Figure 2 (c), to select the rules corresponding to *r3*, *r5* and *r6*, the highest value for the support threshold is roughly 0.17, while for the other two rules, it is 0.49. If both the confidence threshold and the support threshold are appropriately selected, the most possible association rules will be distinguished from others. As a result, the five prerequisite relations can be discovered in this experiment.

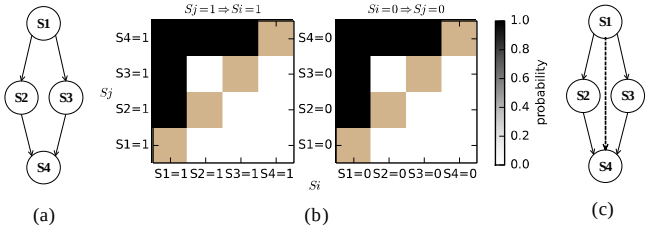


Figure 3. (a) Presupposed prerequisite structure of the skills in the simulated data; (b) Probabilities of the association rules in the simulated data given *minconf*=0.76 and *minsup*=0.125, brown squares denoting impossible rules; (c) Discovered prerequisite structure

Figure 3 (b) illustrates the probabilities of the corresponding association rules in the simulated data given *minconf*=0.76 and *minsup*=0.125. A square's color indicates the probability of the corresponding rule. Five association rules in each of the figures whose probabilities are almost 1.0 are deemed to exist. And the prerequisite relations corresponding to the discovered rules are deemed to exist. To qualitatively construct the prerequisite structure of skills, every discovered prerequisite relation is represented by an arc. It should be noted that the arc representing

the relation that *S1* is a prerequisite of *S4* is not present in Figure 3 (a) due to the transitivity of prerequisite relation. Consequently, the prerequisite structure discovered by our method which is shown in Figure 3 (c), is completely in accordance with the presupposed structure shown in Figure 3 (a).

4.2 Real Testing Data

Data set. The ECPE (Examination for the Certification of Proficiency in English) data set is available through the R package CDM [21], which comes from a test developed and scored by the English Language Institute of the University of Michigan [22]. A sample of 2933 examinees is tested by 28 items on 3 skills, i.e. Morphosyntactic rules (*S1*), Cohesive rules (*S2*), and Lexical rules (*S3*). The parameter estimation tool in the R package CDM [21] for DINA model is also used in this experiment to estimate the slip and guess parameters of items according to the student response data. And with the estimated slip and guess parameters, the probabilistic knowledge states of students are assessed according to the DINA model, which are the input data for discovering the prerequisite structure of skills.

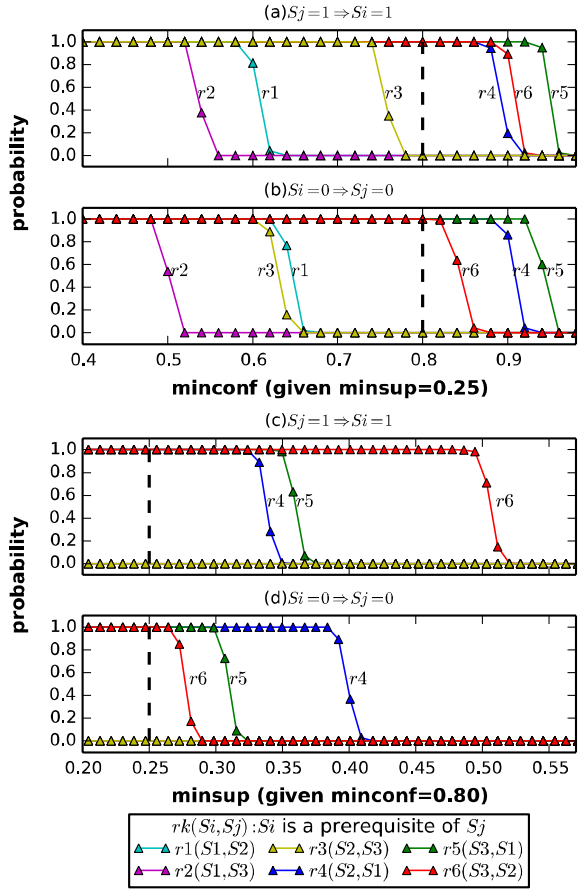


Figure 4. The probabilities of the association rules in the ECPE data given different confidence or support thresholds

Result. The effect of different confidence thresholds on the association rules in the ECPE data is depicted in Figure 4 (a) and (b) given the support threshold as a constant (0.25 here). In each figure, there are three association rules that can satisfy a significantly higher confidence threshold than others. The maximum value of the confidence threshold for them is roughly 0.82. And these rules in the two figures correspond to the same set of prerequisite relation candidates, that is, *r4*, *r5* and *r6*. Thus these candidates are most likely to exist. It can be noticed that in Figure 4 (a) the rule *S3*=1⇒*S2*=1 can satisfy a relatively high confidence threshold. The maximum threshold value that it can satisfy is roughly 0.74. However, its counterpart in Fig 4 (b), i.e. the rule *S2*=0⇒*S3*=0, cannot satisfy a confidence threshold higher than 0.6. When a strong prerequisite relation is required, the relation corresponding to the two rules cannot be selected. Only when both the two types of rules can satisfy a high confidence, the corresponding prerequisite relation is considered strong. Likewise, the effect of different support thresholds is shown in Figure 4 (c) and (d), where the confidence threshold is given as 0.80. And in each figure, only the three association rules which satisfy the confidence threshold are sensitive to different support thresholds. It can also be found that these rules are supported by a considerable proportion of the sample. Even when *minsup*=0.27, all the three rules in each figure satisfy it. According to the figures, when the support and confidence thresholds are appropriately selected, these rules can be distinguished from others. Consequently, the strong prerequisite relations can be discovered.

Given the confidence and support thresholds as 0.80 and 0.25 respectively, for instance, the probabilities of the corresponding association rules are illustrated in Figure 5 (b). The rules that satisfy the two thresholds (with a probability of almost 1.0) are deemed to exist, which are evidently distinguished from the rules that do not (with a probability of almost 0.0). Three prerequisite relations shown in Figure 5 (c) are found in terms of the discovered association rules. To validate the result, we compare it with the findings of another research on the same data set. The attribute hierarchy, namely the prerequisite structure of skills, in ECPE data has been investigated by Templin and Bradshaw [22] as Figure 5 (a). Our discovered prerequisite structure totally agrees with their findings.

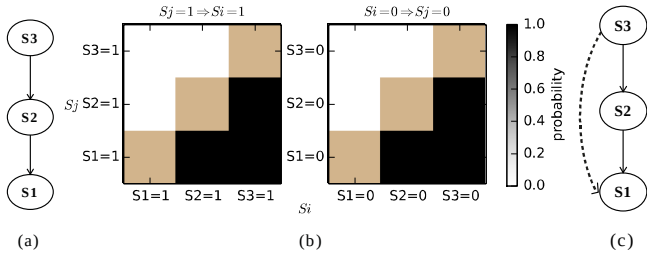


Figure 5. (a) Prerequisite structure of the skills in the ECPE data discovered by Templin and Bradshaw [22]; (b) Probabilities of the association rules in the ECPE data given *minconf*=0.80 and *minsup*=0.25, brown squares denoting impossible rules; (c) Discovered prerequisite structure

4.3 Real Log Data

Data set. We use the 2006-2007 school year data of the curriculum "Bridge to Algebra" [23] which incorporates the log files of 1146 students collected by Cognitive Tutor, an ITS for mathematics learning. The units in this curriculum involve distinct mathematical topics, while the sections in each unit involve distinct skills on the unit topic. A set of word problems is provided for each section skill. We use the sections in the units "equivalent fractions" and "fraction operations" as the skills (see Table 2). There are 560 students in the data set performing to learn one or several of the item-type skills in these units. The five skills discussed in our experiment are instructed in the given order in Table 2. A student's knowledge of the prior skills has the potential to affect his learning of the new skill. Hence, it makes sense to estimate whether a skill trained prior to the new skill is a

prerequisite of it. If the prior skill *Si* is a prerequisite of skill *Sj*, students who have mastered skill *Sj* quite likely have previously mastered skill *Si*, and students not mastering the skill *Si* quite likely learn the skill *Sj* with great difficulty. Thus if both the rules *Sj*=1⇒*Si*=1 and *Si*=0⇒*Sj*=0 exist in the data, the prior skill *Si* is deemed a prerequisite of skill *Sj*.

Table 2. Skills in the curriculum "Bridge to Algebra"

Skill	Example
S1: Writing equivalent fractions	Fill in the blank: $\frac{2}{3}=\frac{\square}{6}$.
S2: Simplifying fractions	Write the fraction in simplest form: $\frac{24}{30}=\_$.
S3: Comparing and ordering fractions	Compare the fractions $\frac{3}{4}$ and $\frac{5}{6}$.
S4: Adding and subtracting fractions with like denominators	$\frac{2}{10}+\frac{3}{10}=$
S5: Adding and subtracting fractions with unlike denominators	$\frac{2}{3}-\frac{1}{4}=$

To discover the prerequisite relations between skills, firstly we need to estimate the outcomes of student learning according to the log data. A student learns a skill by solving a set of problems that requires applying that skill. At each opportunity, student knowledge of a skill probably transitions from the unlearned to learned state. Thus their knowledge should be updated each time they go through a problem. The BKT model has been widely used to track the dynamic knowledge states of students according to their activities on ITSs. In the standard BKT, four parameters are specified for each skill [14]: $P(L_0)$ denoting the initial probability of knowing the skill a priori, $P(T)$ denoting the probability of student's knowledge of the skill transitioning from the unlearned to the learned state, $P(S)$ and $P(G)$ denoting the probabilities of slipping and guessing when applying the skill. We implemented the BKT model by using the Bayes Net Toolbox for Student modeling [24]. The parameter $P(L_0)$ is initialized to 0.5 while the other three parameters are initialized to 0.1. The four parameters are estimated according to the log data of students, and the probability of a skill to be mastered by a student is estimated each time the student performs to solve a problem on that skill. In the log data, students learned the section skills one by one and no student relearned a prior section skill. If a prior skill *Si* is a prerequisite of skill *Sj*, the knowledge state of *Si* after the last opportunity of learning it has an impact on learning *Sj*. We use the probabilities about students' final knowledge state of *Si* and *Sj* to analyze whether a prerequisite relation exists between them. Thus students' final knowledge states on each skill are used as the input data of our method.

Result. The probabilities of the association rules in the log data changing with different confidence thresholds are illustrated in Figure 6 (a) and (b) given the support threshold as a small constant (0.05 here). In Figure 6 (a), compared with the rules *S4*=1⇒*S3*=1 and *S5*=1⇒*S3*=1, all the other association rules can satisfy a significantly higher confidence, while in Figure 6 (b) if given *minconf*=0.6, only three rules satisfy it. The effect of different support thresholds on the probabilities of the association rules is depicted in Figure 6 (c) and (d) given the confidence threshold as a constant (0.3 here). All the association rules satisfy the confidence threshold as the probabilities of the rules are almost 1.0 at first. In Figure 6 (c), there are six rules that can satisfy a relatively higher support threshold (e.g. *minsup*=0.2). But in Figure 6 (d), even given *minsup*=0.14, only the rule *S4*=0⇒*S5*=0 satisfy it, and the maximum value for the support threshold that all the rules can satisfy is roughly 0.07.

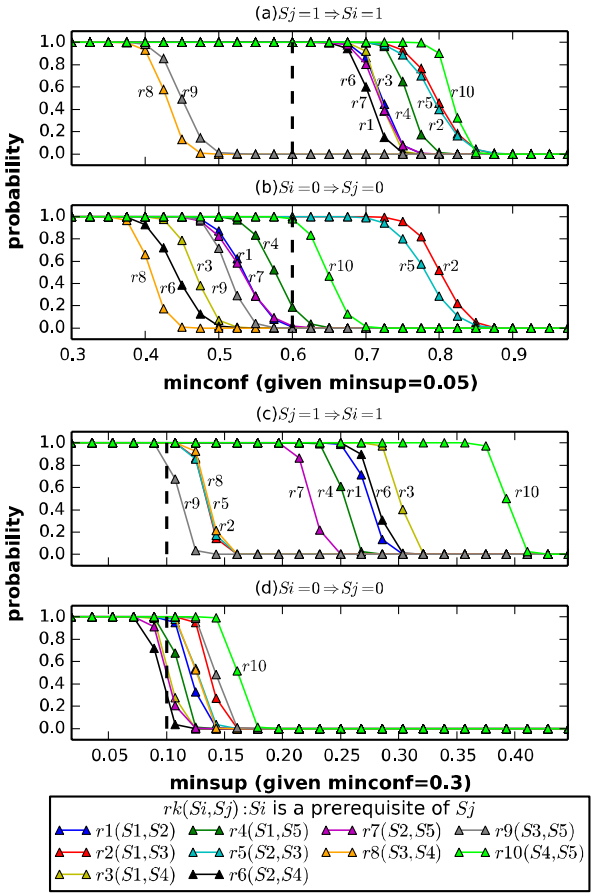


Figure 6. The Probabilities of the association rules in the "Bridge to Algebra 2006-2007" data given different confidence or support thresholds

Given the confidence and support thresholds as 0.6 and 0.1 respectively, the probabilities of the association rules in the log data are depicted in Figure 7 (b). There are eight of the rules in the form of *Sj*=1⇒*Si*=1 (left) and three of the rules in the form of *Si*=0⇒*Sj*=0 (right) discovered, whose probabilities to satisfy the thresholds are almost 1.0. According to the result, only the three prerequisite relations shown in Figure 7 (c), whose corresponding rules both are discovered, are deemed to exist. Figure 7 (a) shows the prerequisite structure of the five skills from the human experts' opinions. It makes sense that the skills *S1* and *S2* rather than skill *S3* are required for learning the skills *S4* and *S5*. This is supported by the chapter warm-up content in the student textbook of the course [25]. The discovered rules in the form of *Sj*=1⇒*Si*=1 completely agree with the structure from human expertise. But the discovered rules in the form of *Si*=0⇒*Sj*=0 is inconsistent with it. The counterparts of a large part of the discovered rules *Sj*=1⇒*Si*=1 do not satisfy the confidence threshold. Even reducing the confidence threshold to the lowest value, i.e. 0.5, the rules *S1*=0⇒*S4*=0 and *S2*=0⇒*S4*=0 still do not satisfy it (see Figure 6 (b)). It seems that the rules *Sj*=1⇒*Si*=1 are more reliable than

Si=0⇒Sj=0 since most of the former can satisfy a higher support threshold than the latter (see Figure 6 (c) and (d)). In addition, the log data is very likely to contain much noise. It is possible that some skills could be learned if students take sufficient training, even though some prerequisites are not previously mastered. In this case, the support count $\sigma(Si=0, Sj=1)$ would increase. Or perhaps students learned the prerequisite skills by solving the scaffolding questions in the process of learning new skills, even though they performed not mastering the prerequisite skills before. In this case, the observed values of $\sigma(Si=0, Sj=1)$ would be higher than the real values. According to the equations (4) and (5), if $\sigma(Si=0, Sj=1)$ increases, the confidence of the rules will decrease. And when the noise appears in the data, the confidences of the association rules which are supported by a small proportion of sample will be affected much more than those supported by a large proportion of sample.

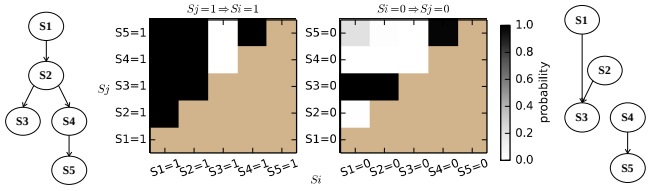


Figure 7. (a) Prerequisite structure from human expertise; (b) Probabilities of the association rules in the "Bridge to Algebra 2006-2007" data given *minconf*=0.6 and *minsup*=0.1, brown squares denoting impossible rules; (c) Discovered prerequisite structure

4.4 Joint Effect of thresholds

We have discussed the effect of one threshold on the probability of association rules while eliminating the effect of the other one in the three experiments. To determine the values for the thresholds, we investigate how the two thresholds simultaneously affect the probability of an association rule. Figure 8 depicts how the probabilities of the association rules for the skill pair *S2* and *S3* in the ECPE data change with different support and confidence thresholds, where (a) and (c) involve one relation candidate while (b) and (d) involve the other one. The figures demonstrate that the probability of a rule decreases almost from 1.0 to 0.0 when the confidence and support thresholds vary from low to high. It can be found that the rules in the left figures can satisfy an evidently higher confidence threshold than those in the right figures, and have the same support distributions with them. If we set *minconf*=0.8 and *minsup*=0.25, only the rules in the left figures satisfy them. Suppose that a rule satisfy the thresholds if its probability is higher than 0.95, i.e. *minprob*=0.95. When we change the values of the confidence and support thresholds from 0.0 to 1.0, for each rule, we can find a point whose coordinates consist of the maximum values of the confidence and support thresholds that the rule can satisfy. Finding the optimal point is hard and there are probably several feasible points. To simplify the computation, the thresholds are given by a sequence of discrete values from 0.0 to 1.0. We find the maximum value for each threshold when only one threshold affects the probability of the rule given the other as 0.0. And for each threshold, *minprob* is given as 0.97, roughly the square root of the original value. The found maximum values for the two thresholds are the coordinates of the point. The found point is actually an approximately optimal point. For convenience, the point is named maximum threshold point in this paper. The points for all the rules in the three data sets are found by our method as well as plotted in Figure 9 (some points overlap). When we set certain values to the thresholds, the points located in the upper right area satisfy them and the related rules are deemed to exist. For one prerequisite relation, a couple of related points should be verified. Only when both of them are located in the upper right area, they are considered eligible to uncover the prerequisite relation. The eligible points in Figure 8 and Figure 9 are indicated given the thresholds.

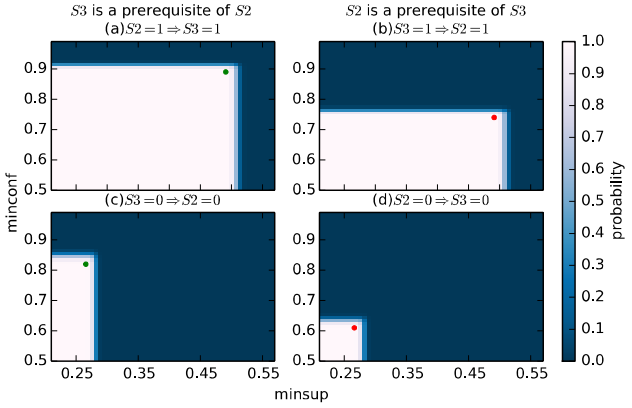


Figure 8. Probabilities of the association rules within the skill pair *S2* and *S3* in the ECPE data given different confidence and support thresholds, and their maximum threshold points which are eligible (green) or not (red) given *minconf*=0.8 and *minsup*=0.25

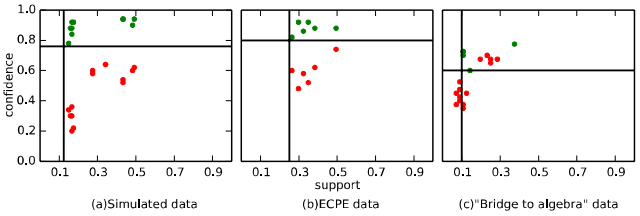


Figure 9. Maximum threshold points for the association rules in our three experiments, where eligible points are indicated in green given the thresholds

5. CONCLUSION AND DISCUSSION

Discovering the prerequisite structure of skills from data is challenging in domain modeling since skills are the latent variables. In this paper, we propose to apply the probabilistic association rules mining technique to discover the prerequisite structure of skills from student performance data. Student performance data is preprocessed by an evidence model. And then the probabilistic knowledge states of students estimated by the evidence model are used as the input data of probabilistic association rules mining. Prerequisite relations between skills are discovered by estimating the corresponding association rules in the probabilistic database. The confidence condition of an association rule in our method is similar to the statistical hypotheses used in the POKS algorithm for determining the prerequisite relations between observable variables (see the details in [5]). But our method targets on the challenge of discovering the prerequisite relations between latent variables from the noisy observable data. In addition, our method takes the coverage into account (i.e. the support condition), which could strengthen the reliability of the discovered prerequisite relations. Determining the appropriate confidence and support thresholds is a crucial issue in our method. The effect of a single threshold and the joint effect of two thresholds on the probabilities of the rules are

discussed. The maximum threshold points of the probabilistic association rules are proposed for determining the thresholds. We adapt our method to two common types of data, the testing data and the log data, which are preprocessed by different evidence models, the DINA model and the BKT model. An accurate Q-matrix is required for the evidence models, which is a limitation of our method. According to the results of the experiments in this paper, our method performs well to discover the prerequisite structures from a simulated testing data set and a real testing data set. However, applying our method in the log data still needs to be improved. Since much noise exist in the log data, the strategies to reduce the noise need to be applied. The prerequisite structures of skills discovered by our method can be applied to assist human experts in skill modeling or to validate the prerequisite structures of skills from human expertise.

6. REFERENCES

[1] Käser, T., Klinger, S., Schwing, G., Gross, M.: Beyond Knowledge Tracing: Modeling Skill Topologies with Bayesian Networks. In *Proceedings of the 12th International Conference on Intelligent Tutoring Systems*, Honolulu, USA, 188-198, 2014

[2] Chen, Y., Wuillemin, P.H., Labat, J.M.: Bayesian Student Modeling Improved by Diagnostic Items. In *Proceedings of the 12th International Conference on Intelligent Tutoring Systems*, Honolulu, USA, 144-149, 2014

[3] Falmagne, J.C., Cosyn, E., Doignon, J.P., Thiéry, N.: The Assessment of Knowledge, in Theory and in Practice. In *Proceedings of the 4th International Conference on Formal Concept Analysis*, Dresden, Germany, 61-79, 2006

[4] Heller, J., Steiner, C., Hockemeyer, C., Albert, D.: Competence-based Knowledge Structures for Personalized Learning. *International Journal on E-learning*, 5, 75-88, 2006

[5] Desmarais, M.C., Meshkinfam, P., Gagnon, M.: Learned Student Models with Item to Item Knowledge Structures. *User Modeling and User-adapted Interaction*, 16(5), 403-434, 2006

[6] Spirtes, P., Glymour, C., Scheines, R.: *Causation, Prediction, and Search*. 2nd Edition, The MIT Press, Cambridge, 2000

[7] Pavlik Jr., P.I., Cen, H., Wu, L., Koedinger, K.R.: Using Item-type Performance Covariance to Improve the Skill Model of an Existing Tutor. In *Proceedings of the 1st International Conference on Educational Data Mining*, Montreal, Canada, 77-86, 2008

[8] Vuong, A., Nixon, T., Towle, B.: A Method for Finding Prerequisites within a Curriculum. In *Proceedings of the 4th International Conference on Educational Data Mining*, Eindhoven, Netherlands, 211-216, 2011

[9] Tseng, S.S., Sue, P.C., Su, J.M., Weng, J.F., Tsai, W.N.: A New Approach for Constructing the Concept Map. *Computers & Education*, 49(3), 691-707, 2007

[10] Brunskill, E.: Estimating Prerequisite Structure from Noisy Data. In *Proceedings of the 4th International Conference on Educational Data Mining*, Eindhoven, Netherlands, 217-222, 2011

[11] Scheines, R., Silver, E., Goldin, I.: Discovering Prerequisite Relationships among Knowledge Components. In *Proceedings of the 7th International Conference on Educational Data Mining*, London, UK, 355-356, 2014

[12] Agrawal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, USA, 487-499, 1994

[13] Roussos, L.A., Templin, J.L., Henson, R.A.: Skills Diagnosis Using IRT-based Latent Class Models. *Journal of Educational Measurement*, 44(4), 293-311, 2007

[14] Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253-278, 1995

[15] Barnes, T.: The Q-matrix Method: Mining Student Response Data for Knowledge. *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005

[16] Desmarais, M.C., Naceur, R.: A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-Matrices. In *Proceedings of the 16th International Conference on Artificial Intelligence in Education*, Memphis, USA, 441-450, 2013

[17] González-Brenes, J.P.: Modeling Skill Acquisition Over Time with Sequence and Topic Modeling. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, San Diego, USA, 296-305, 2015

[18] Bernecker, T., Kriegel, H.P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic Frequent Itemset Mining in Uncertain Databases. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 119-128, 2009

[19] Chui, C.K., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In *Proceedings of the 11th PAKDD Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Nanjing, China, 47-58, 2007

[20] Sun, L., Cheng, R., Cheung, D.W., Cheng, J.: Mining Uncertain Data with Probabilistic Guarantees. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 273-282, 2010

[21] Robitzsch, A., Kiefer, T., George, A.C., Uenlue, A.: Package CDM (Version 3.4-21, 2014), http://cran.r-project.org/web/packages/CDM/index.html

[22] Templin, J., Bradshaw, L.: Hierarchical Diagnostic Classification Models: A Family of Models for Estimating and Testing Attribute Hierarchies. *Psychometrika*, 79, 317-339, 2014

[23] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., & Koedinger, K.R.: Bridge to Algebra 2006-2007. Development data set from KDD Cup 2010 Educational Data Mining Challenge. Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

[24] Chang, K., Beck, J., Mostow, J., & Corbett, A.: A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan, 104-113, 2006

[25] Hadley, W.S., Raith, M.L.: *Bridge to Algebra Student Text*. Carnegie Learning. 2008

Choosing to Interact: Exploring the Relationship Between Learner Personality, Attitudes, and Tutorial Dialogue Participation

Aysu Ezen-Can
Department of Computer Science
North Carolina State University
aezen@ncsu.edu

Kristy Elizabeth Boyer
Department of Computer Science
North Carolina State University
keboyer@ncsu.edu

ABSTRACT

The tremendous effectiveness of intelligent tutoring systems is due in large part to their interactivity. However, when learners are free to choose the extent to which they interact with a tutoring system, not all learners do so actively. This paper examines a study with a natural language tutorial dialogue system for computer science, in which students interacted with the JavaTutor system through natural language dialogue over the course of problem solving. We explore the relationship between students' level of dialogue interaction and learner characteristics including personality profile and pre-existing attitudes toward the learning task. The results show that these learner characteristics are significant predictors of the extent to which students engage in dialogue with the tutoring system, as well as the number of task actions students make. By identifying students who may not engage with tutoring systems as readily, this work constitutes a step toward building adaptive systems that successfully support a variety of students with different attitudes and personalities.

Keywords

Learner characteristics, personality, disengagement, tutorial dialogue

1. INTRODUCTION

Tutorial dialogue systems effectively support learning through rich natural language dialogue [7,8,14,19]. However, the effectiveness of tutorial dialogue systems, like other adaptive learning environments, depends in large part on students' willingness to interact with them [18]. Interaction varies tremendously across individual students and student populations. We observe various types of *disengagement* including lack of motivation or interest for the learning task [10], as well as *gaming* an intelligent tutor by exploiting properties of the learning environment [2,4].

In addition to these factors, individual differences such as self-reported interest in the task and confidence in learning have been found to be strong predictors of engagement [6]. Similarly, students' hidden attitudes toward learning [1] and motivation for the task [3] may be highly influential. Boredom, which is associated with reduced motivation to perform the activity [15], has been positively correlated with attention problems and negatively correlated with performance. Students' participation in tutorial dialogue has also been found to be associated with the students' expectations [11], and in human-human tutorial dialogue, student personality traits have recently been found to be significant factors [16]. However, the field is far from a full understanding of the factors that influence students' choices to engage or interact with tutorial dialogue systems.

This paper presents an investigation into the relationship between student characteristics and interactions with a tutorial dialogue system. We hypothesized that students' personality profile, for example their tendencies toward extraversion or openness, would be significantly associated with the level of natural language interaction observed within a tutorial dialogue system. We also hypothesized that students' attitudes toward the learning task would be a significant factor in their interactions with the system. We examine these hypotheses within a data set of 51 university students interacting with the JavaTutor tutorial dialogue system for introductory computer science. Regression models were built that predict both dialogue and task participation by the students, who have the choice to interact with the dialogue system as little or as much as desired over the course of the learning tasks. The models demonstrate that students' attitudes and personalities are significantly predictive of their willingness to interact with the tutorial dialogue system. The findings suggest that some learner characteristics may put students at risk of low participation with a tutorial dialogue system, and constitute a first step toward proactively adapting the systems to benefit these learners.

2. TUTORING STUDY

The JavaTutor tutorial dialogue system (Figure 1) supports students in solving introductory computer programming problems in the Java programming language while interacting in textual natural language. Students are provided with a series of learning tasks that build on each other to guide the students through creation of a simple text-based adventure game.[1]

The study reported here was conducted with the JavaTutor tutorial dialogue system in 2014. The students (12 female; 39 male; mean age = 21) were drawn from a university-level engineering class. They interacted with the tutorial dialogue system for one session lasting approximately 45 minutes.

[1]Implementation details of the system are beyond the scope of this paper but are described in [9].

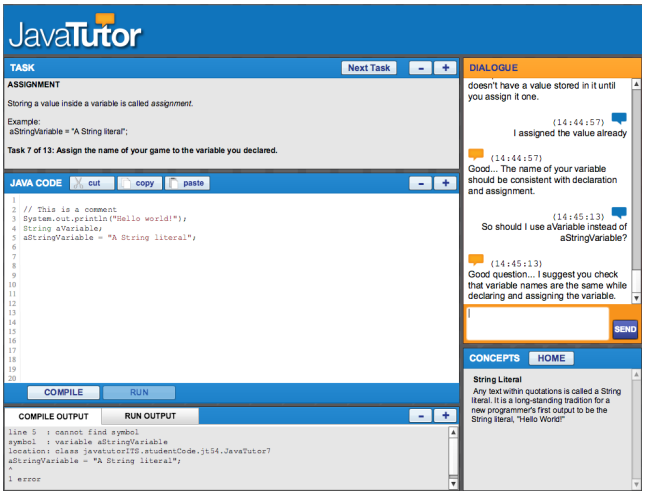


Figure 1: Screenshot from the tutorial dialogue system.

Prior to interacting with JavaTutor, students took a pre-survey that included validated items to measure goal orientation [17], general self-efficacy [5], confidence in learning computer science and programming [13], and personality profile using a concise version of the Big Five Inventory [12]. Students also completed a pre-test and posttest before and after their interaction with JavaTutor.

3. ANALYSIS

Students were instructed that they could make comments, pose questions, and request feedback at any time through textual dialogue. Overall, students interacting with JavaTutor achieved significant learning gains from pre-test to posttest (average= 12%, median= 13.4%, stdev = 32%, $p = 0.001$). However, we observed that 58.8% of students never made an utterance. For students who did engage in dialogue with the tutor, the average number of utterances was 5.1 (stdev=7.36, median= 2). Regardless of the extent to which they chose to engage in natural language dialogue, all students received some tutorial dialogue utterances based upon the system's model of feedback for task events.

Our goal is to identify the factors that may be influential in students' levels of interaction with the system. To this end, we built multiple regression models. The remainder of this section describes the analysis.

3.1 Response Variables

Based upon the logged interaction traces, we extracted dialogue and task events and used them to compute a numeric representation of the student's level of interaction with the system. For dialogue interaction we utilized the *number of utterances* written by each student. The range of number of student utterances was between 0 and 33.

We extracted four features that represent interaction of students with the system throughout tutoring. The first of these four features is *number of content changes* which refers to the changes in the student's programming code, as the code they write is referred to as content pane. We also computed *the number of compile events* and *number of run activities*. The number of compile/run events ranges from 4 to 224, whereas the number of content changes ranges from 88 to 1099 to complete the series of learning tasks.

Finally, we computed the *number of tutor messages* each student received. The tutoring systems provided students with feedback. The number of messages received is closely related to the number of actions that triggered tutor feedback, which is also a measure of participation. The minimum number of tutor messages provided to any student was 8, whereas the largest number of tutor messages to a student during a tutoring session was 121. We built separate multiple regression models to predict level of dialogue interaction and level of task interaction.

3.2 Predictor Variables

We hypothesized that several learner characteristics were significantly associated with level of interaction in the system. We provided these variables for selection within the models (see Table 1). All of the predictors were standardized to a common scale before model building.

Predictor variable	Example survey item/ Description
Computer science confidence	*I am sure that I can learn programming.*
Perceived computer science usefulness	*I'll need programming for my future work.*
Motivation toward computer science	*Programming is enjoyable and stimulating to me.*
General self-efficacy	*I will be able to achieve most of the goals that I have set for myself.*
Learning goal orientation	*I often look for opportunities to develop new skills and knowledge.*
Performance demonstration	*I like to show that I can perform better than my coworkers.*
Failure avoidance	*Avoiding a show of low ability is more important to me than learning a new skill.*
Achievement goals	*It is important for me to do better than other students.*
Gender	Male/female
Age	Age of the student
University class standing	The year that the student is in the university
Perception of student's own computer skill	*How skilled are you with computers, compared to the average person?*
Extraversion	*I see myself as someone who is talkative.*
Agreeableness	*I see myself as someone who is helpful and unselfish with others.*
Conscientiousness	*I see myself as someone who does a thorough job.*
Neuroticism	*I see myself as someone who is depressed, blue.*
Openness	*I see myself as someone who is original, comes up with new ideas.*
Pre-test score	Score showing the performance of the student before tutoring session

Table 1: Predictor variables from pre-survey and pre-test.

3.3 Modeling Level of Participation

We built separate models for each of the response variables (number of utterances, compile/run events, content changes, received tutor messages). For each response variable we used the whole dataset

and selected features via stepwise linear regression. Because the goal was to investigate relationships between pre-measures (student characteristics, attitudes) and level of participation, we conducted descriptive analyses using the entire data set for model building.

The model for number of dialogue utterances (Table 2) revealed that students' failure avoidance characteristic is a significant predictor of tutorial dialogue interactivity. Students who indicated that they tend to avoid tasks in which they may have higher chance of failure wrote fewer utterances to the system.

Number of utterances =	**Coefficient**	***p***
Failure Avoidance	-0.3089	0.0274
~1 (intercept)		1
RMSE = 0.961		
R^2 = 0.0954		

Table 2: Stepwise linear regression model for the number of utterances.

The model for number of compile/run events during tutoring session showed that students' personality scores, particularly the binary agreeableness score, was a significant predictor of participation from a task-related perspective. The students who were more agreeable (indicated as a 1 for the model, rather than a 0) made more task interactions considering compile/run events as shown in Table 3. The other regression model having the number of content changes as a response variable did not produce significant results.

Number of compile/run =	**Coefficient**	***p***
Agreeableness (binarized)	0.2897	0.0392
~1 (intercept)		1
RMSE = 0.967		
R^2 = 0.0839		

Table 3: Stepwise linear regression model for number of compile/run events.

Another regression model that showed significant results was the regression model that predicted the number of tutor messages students received. Interestingly, both student perceptions (computer science confidence and motivation) and personality (openness score from Big Five Inventory) were selected by the model as shown in Table 4. There was a negative correlation between computer science confidence and tutor messages, however it was the opposite for computer science motivation. The students who were more motivated to study computer science interacted more with the system, triggering more tutor messages. Also, the students who had low confidence towards programming received less tutor feedback. Figure 2 shows the scatter plots for both computer science motivation and confidence measures.

Discussion. Understanding how student characteristics are associated with tutorial dialogue interaction holds great promise for identifying possible disengagement types and taking adaptive action during tutoring sessions to further improve learning effectiveness. The results of the models indicate that as hypothesized, student characteristics such as personality profile were significantly predictive of the student's level of interactivity with the tutorial dialogue system. We found that students' attitudes and personalities have significant relationships with their level of participation in terms of both dialogue and task.

Number of tutor messages =	**Coefficient**	***p***
Age	0.3802	0.0033
Computer science confidence * Openness	-0.5244	0.0008
Computer science motivation Openness	0.5317	0.00006
~1 (intercept)		1
RMSE = 0.739		
R^2 = 0.52		

Table 4: Stepwise linear regression model for number of tutor messages received.

Another important finding was that although pre-test was present in all regression models as an independent variable, it was not significantly predictive of either the number of utterances or the task activities. In other words, the level of participation was more correlated to student characteristics than to their incoming knowledge. These results are important for understanding how to better foster interaction with intelligent tutoring systems. If we can identify students who tend to participate less or become disengaged, the system can automatically adapt to these students with scaffolding. For instance, when a student with low motivation toward the task is identified, the tutorial dialogue system might put particular emphasis on moves that are part of "adjacency pairs," such as asking a question and awaiting a student response. Adapting the task may also be appropriate in these cases. By utilizing information that we can glean from quick pre-measures, we may be able to significantly improve the effectiveness of the system.

4. CONCLUSION

Adapting to broader populations with varying characteristics is crucial for increasing the use of intelligent tutoring systems and making them more effective. A central challenge is determining the factors that might affect level of participation with intelligent systems. The current literature is far from totally understanding underlying relationships between student characteristics and how they affect system interactions during tutoring. The findings presented here have identified student characteristics such as level of failure avoidance which are particularly strongly associated with low interaction.

Several directions of future work are promising. First, incorporating multiple sources of information such as multimodal features (e.g., posture, gesture, eye gaze) can help us better understand students and respond in real time to engage them in more interactions. Each of these types of features has been shown to contribute to modeling student behavior. Additionally, customizing scaffolding to different learner characteristics is very promising. Modifying the realized utterances delivered to students based on their personality style, gender, and skill are likely to improve interactions with the system. It is important to devise and investigate strategies for learners of all characteristics in order to better engage students and help them learn more.

Acknowledgments The authors wish to thank the members of the LearnDialogue group at North Carolina State University for their helpful input. This work is supported in part by the Department of Computer Science at North Carolina State University and the National Science Foundation through Grants IIS-1409639, and the

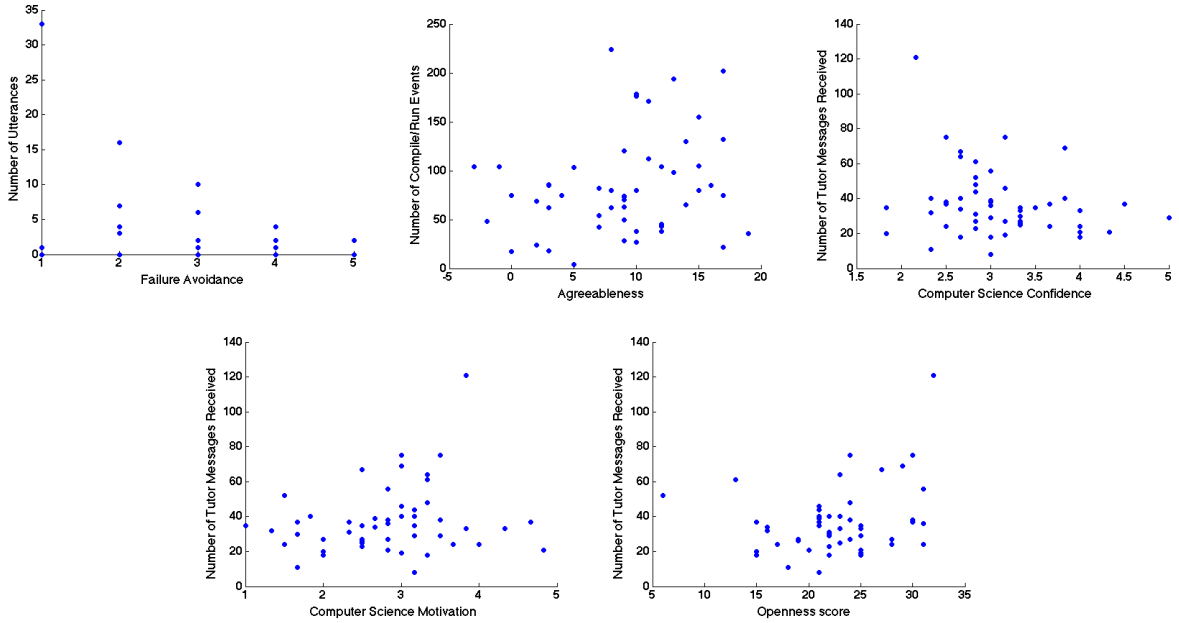


Figure 2: Scatter plots of various predictors and response variables.

STARS Alliance, CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

5. REFERENCES

[1] I. Arroyo and B. P. Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *Proceedings of AIED*, pages 33–40, 2005.

[2] R. S. Baker, A. de Carvalho, J. Raspat, V. Aleven, A. T. Corbett, and K. R. Koedinger. Educational software features that encourage and discourage "gaming the system". In *Proceedings of AIED*, pages 475–482, 2009.

[3] C. R. Beal, L. Qu, and H. Lee. Mathematics motivation and achievement as predictors of high school students' guessing and help-seeking with instructional software. *Journal of Computer Assisted Learning*, 24(6):507–514, 2008.

[4] J. E. Beck. Engagement tracing: Using response times to model student disengagement. In *Proceedings of AIED*, pages 88–95, 2005.

[5] G. Chen, S. M. Gully, and D. Eden. Validation of a new general self-efficacy scale. *Organizational research methods*, 4(1):62–83, 2001.

[6] S. D'Mello, C. Williams, P. Hays, and A. Olney. Individual differences as predictors of learning and engagement. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pages 308–313, 2009.

[7] S. K. D'Mello, B. Lehman, and A. Graesser. A motivationally supportive affect-sensitive AutoTutor. In *New perspectives on affect and learning technologies*, pages 113–126. 2011.

[8] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, and G. Campbell. BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *IJAIED*, 24(3):284–332, 2014.

[9] A. Ezen-Can and K. E. Boyer. A tutorial dialogue system for real-time evaluation of unsupervised dialogue act classifiers: Exploring system outcomes. *To appear*.

[10] K. Forbes-Riley and D. Litman. When does disengagement correlate with performance in spoken dialog computer tutoring? *IJAIED*, 22(2):19–41, 2008.

[11] G. T. Jackson, A. C. Graesser, and D. S. McNamara. What students expect may have more impact than what they know or feel. In *Proceedings of AIED*, pages 73–80, 2009.

[12] O. P. John, L. P. Naumann, and C. J. Soto. Paradigm shift to the integrative Big Five trait taxonomy. *Handbook of personality: Theory and research*, 3:114–158, 2008.

[13] C. Lee and P. Bobko. Self-efficacy beliefs: Comparison of five measures. *Journal of Applied Psychology*, 79(3):364, 1994.

[14] D. Litman and S. Silliman. ITSPOKE : An Intelligent Tutoring Spoken Dialogue System. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8, 2004.

[15] R. Pekrun, T. Goetz, L. M. Daniels, R. H. Stupnisky, and R. P. Perry. Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology*, 102(3):531, 2010.

[16] A. K. Vail and K. E. Boyer. Adapting to Personality Over Time: Examining the Effectiveness of Dialogue Policy Progressions in Task-Oriented Interaction. In *Proceedings of the Annual SIGDIAL Meeting*, pages 41–50, 2014.

[17] D. VandeWalle, W. L. Cron, and J. W. Slocum Jr. The role of goal orientation following performance feedback. *Journal of Applied Psychology*, 86(4):629, 2001.

[18] K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney, and C. P. Rosé. When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1):3–62, 2007.

[19] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of ITS*, pages 158–167, 2002.

Considering the influence of prerequisite performance on wheel spinning

Hao Wan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
hale@wpi.edu

Joseph Barbosa Beck
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
josephbeck@wpi.edu

ABSTRACT

The phenomenon of wheel spinning refers to students attempting to solve problems on a particular skill, but becoming stuck due to an inability to learn the skill. Past research has found that students who do not master a skill quickly tend not to master it at all. One question is why do students wheel spin? A plausible hypothesis is that students become stuck on a skill because they do not understand the necessary prerequisite knowledge, and so are unable to learn the current skill. We analyzed data from the ASSISTments system, and determined the impact of how student performance on prerequisite skills influenced ability to learn postrequisite skills. We found a strong gradient with respect to knowledge of prerequisites: students in the bottom 20% of pre-required knowledge exhibited wheel spinning behavior 50% of the time, while those in the top 20% of pre-required knowledge exhibited wheel spinning behavior only 10% of the time. This information is a statistically reliable predictor, and considering it results in a modest improvement in our ability to detect wheel spinning behaviors: R2 improves from 0.264 to 0.268, and AUC improves from 0.884 to 0.888.

Keywords

Wheel Spinning; Prerequisite; Student Model.

1. INTRODUCTION

Many Intelligence Tutoring Systems (ITS) make use of a mastery learning framework where students continue practicing a skill until they master it. However, some students are unable to achieve mastery despite having numerous opportunities to practice the skill. As a result, these students are stuck in the mastery learning cycle of the ITS and are given additional problems on a topic they are unable to master. We refer to these students as "wheel spinning" on the skill. The term wheel spinning comes from a car that is stuck in snow or mud, and despite rapid movement of the wheels, the car is going nowhere. As defined in [1], a student who takes 10 practice opportunities without mastering a skill is considered to be wheel spinning on this skill. Based on this definition, they also point out that about 31% student-skill pairs in CAT and 38% in ASSISTments are wheel spinning. This earlier work identified the students, but did not provide an explanation for why certain students become stuck. Thus, the next question to address is to understand why students wheel spin in order to provide effective remediation to those students.

Beck and Gong [1] developed a model, consisting of 8 features, to predict which students will wheel spin on a skill. They found that there is a relationship between wheel spinning and gaming the system [12]. Beck and Rodrigo [2] constructed a causal model (using non-Western students) that situated wheel spinning in the face of affective factors. They found that wheel spinning and gaming were strongly related. This work also presented a path model that found gaming was not causal of wheel spinning, but rather, wheel spinning was related to a lack of prior knowledge, which in turn led to gaming. A more concrete wheel spinning model is developed in [3], in which three aspects of features are considered: student in-tutor performance, the seriousness of the learner, and general factors. However, these models do not provide actionable results for how to make a student less likely to wheel spin on a skill, or how to get an already wheel spinning student unstuck.

A natural question is why are some students able to learn a skill and achieve mastery, while other students fail to do so? One plausible hypothesis of what makes wheel-spinning students different from their peers is a difference in ability to learn the skill. Students certainly differ in cognitive abilities, but addressing such would be beyond the scope of most interventions ITS developers can develop. Another plausible difference in ability to learn the skill is due to differences in student preparation. For example, if students do not understand the concept of equivalent fractions, they will have great difficulty mastering the later skill of addition of fractions, which requires them to solve problems such as 1/3 + 1/4.

We define a skill S's prerequisite skills as those skills necessary to be mastered before studying skill S. This prerequisite structure has been used to improve different student models in many research works. For example, Carmona et al. [4] add a new prerequisite layer into student model based on Bayesian Networks. Their experiments suggest that the prerequisite relationships can improve the model's efficiency in diagnosing students. Botelho et al. use prerequisite structure to estimate students' initial knowledge for subsequent skills [5].

Therefore, in this paper, we incorporate the prerequisite structure into wheel spinning model, in order to check if prerequisite performance has impact in wheel spinning of post-skills. Although prior research has proposed automatic algorithms of adapting prerequisite structures [6] [7] [8], we instead use a prerequisite structure developed by a domain expert.

As an overview, we abstract students' prerequisite performance as a feature, and then add this feature into the wheel-spinning model [1]. Our main points include: 1) determine if there is connection

between the prerequisite performance and the wheel spinning of post-skill; 2) explore how prerequisite factor would affect wheel spinning model; 3) compare the prerequisite factor with another possible effect that could cause wheel spinning – students' general learning ability. The rest paper is organized as following: Section 2 describes the wheel-spinning model; Section 3 introduces our method of how to represent prerequisite performance; results are shown in Section 4, and further discussion is in Section 5; conclusion and future works are made in Section 6.

2. WHEEL SPINNING MODEL

The wheel spinning model used in this work is mainly derived from the one in [1], but there are two differences between them, we will explain later. This model is fitted using logistic regression algorithm in SPSS on the following features:

a) The number of prior correct responses by the student on this skill. This feature is proved useful in the Performance Factors Analysis model (PFA) [9].
b) The number of problems in a row correctly responded by the on the skill prior to the current problem. Since for this paper we are operationalizing mastery as 3 correct responses in a row[1], the number of consecutive correct responses is an important factor. The value of this feature is from 0 to 2.
c) The exponential mean Z-score of response times on this skill. The response time for each item is transferred into a Z-score, and then exponential mean is calculated for each student by: $\gamma * \text{prior\_average} + (1-\gamma) * \text{new\_observation}$, with $\gamma = 0.7$ found to work well in practice in prior research, and so we have retained it here.
d) The exponential mean count of rapid guessing. This measures how often the student was rapidly guessing.
e) The exponential mean count of rapid response. This measures how often the student took a rapid response. This feature as well as the feature (d) reflects how serious the student is learning the skill through the tutoring system. Similar features related with "gaming" the system were used in gaming detectors as in [10] [11] [12].
f) Count of bottom-out hint. The number of times the student reached a bottom-out hint on this skill prior to the current problem.
g) The exponential mean count of 3 consecutive bottom-out-hints. This measures how often the student reached bottom out hints on 3 consecutive problems.
h) Skill identification.
i) Prior response count.

As aforementioned, the model in our experiments is different from the Beck and Gong's model [1] in two places: one is that we use one more feature in the model, the feature b) above; the other is that in some experiments, we treat the last feature – prior response count – as a covariate, not a factor like in their model. We found this parameter's affect was approximately linear, and thus treating it as a covariate made more sense. We call the model based on these 9 features the baseline model, and compare it with a model that includes the prerequisite performance.

3. METHOD

3.1 Computing Students' Performance on Skills

In this paper, our goal is to find the influence of students' prerequisite performance on wheel spinning. So the first step is to choose which measure to represent students' performance on each skill. In this work, we regard a student's percentage of correct responses to questions involving a skill to be his performance on that skill.

However, a student could answer correctly, by chance, even though this student does not understand the skill at all. Similarly, a student could give the wrong answer through a careless mistake, as in the guess and slip parameters in the Knowledge Tracing model [13]. These two cases will deviate the student's performance from his/her "true understanding" on the skill, especially if the student has very few practices. To deal with these cases, we balance the "accidental performance" with student's overall performance on all skill. The formula for calculating a student's performance on a skill i is:

$$P_i = \frac{1}{2^x} * \overline{R} * S_i + \left(1 - \frac{1}{2^x}\right) * C_i$$

- x: The number of practices on this skill;
- S_i: The percent correctness of skill i, $S_i = \frac{\#\text{correct practices}}{\#\text{overall practices}}$ (over all students). This also reflects the hardness of skill S_i.
- C_i : The student's percent correctness on skill i , $C_i = \frac{\#\text{correct practices}}{\#\text{overall practices}}$ (over the student st_1).
- $R_i = \frac{C_i}{S_i}$: This represents how well the student st_1 does on skill i comparing with the other students.
- $\overline{R} = \frac{\sum_{i=1}^{m} R_i}{m}$: m is the number of the student's started skills.

Table 1. A small sample of students' practices.

Student	Skill	Problem	Correct?
st1	s1	p1	1
st1	s1	p2	0
st1	s2	p3	1
st1	s3	p4	0
st2	s1	p1	1
st2	s1	p2	1
st2	s3	p5	1

Table 2. Calculated skills' hardness and students' performance according to the data in Table 1.

Skill	Correctness	Student performance		Normalized performance	
		st1	st2	st1	st2
s1	0.75	0.48	1.06	0.45	1
s2	1.0	0.78	1.67	0.47	1
s3	0.5	0.28	0.92	0.3	1

[1] We use this definition for consistency with prior work, and for ease of application across systems. This mastery criterion is fairly weak, and presumably underestimates the amount of wheel spinning.

Notice in the formula, the more practices on a skill, the more weight is assigned to the performance on this skill. Take the data in Table 1 as an example. There are in total 4 trials for skill s1, of which 3 are answered correctly, so its correctness is 0.75. The correctness of the other two skills is: s2, 1.0; s3, 0.5. The student, st1, answered two problems of s1, getting one correct and the other incorrect. So this student's correctness of s1 is 0.5, and $R_1(st1) = \frac{0.5}{0.75} = 0.67$. We can also get that $R_2(st1) = 1.0, R_3(st1) = 0$, then $\overline{R}(st1) = 0.56$. Hence, the student st1's estimated understanding on the skill s1 is: $\frac{1}{2^2} * 0.56 * 0.75 + \left(1 - \frac{1}{2^2}\right) * 0.5 = 0.48$. All the performance results are shown in Table 2. Sometimes, a student's adjusted performance is larger than 1, as the student st2's performances on skill s1 and s2. This effect can occur by a student doing very well on a very difficult skill. In this paper, we normalize the values to bring them in the range from 0 to 1.

3.2 Computing Prerequisite Performance

Once the normalized students' performances have been computed, the next step is to think about how to represent prerequisite performances, and then incorporate it into the wheel-spinning model. If a skill has only one pre-required skill, such a representation is straightforward: the student's adjusted performance on that pre-required skill. But what if a skill has multiple prerequisites? In our data set, 39 out of 128 skills have multiple prerequisites. There are a variety of approaches for handling multiple prerequisites. We chose two different methods to compute the prerequisite performance: weakest link and weighted by hardness.

3.2.1 Weakest Link

This method is based on an assumption that learning a skill requires mastery of all its prerequisites. For example, lack knowledge of square or square root might not solve the Pythagorean equation. Therefore, this method regards the prerequisite skill with the worst performance, called weakest link, as the bottom boundary of estimation of prerequisite knowledge.

In this paper, we use the lowest performance value in all prerequisite skills as the wheel-spinning model's input for prerequisite performance. For example, in Table 1, if skill s1's prerequisite skills are s2 and s3, then the prerequisite performance for student st1 on skill s1 is estimated as 0.3 (normalized).

3.2.2 Weighted by Hardness

This method assumes each prerequisite skill has different importance in affecting learning a post-skill, and this importance is determined by how hard the prerequisite skill is. Thus, we sum up a student's prerequisite performances by assigning a corresponding weight to each prerequisite skill, according to the skill hardness. Here we define a skill's hardness to be $1/correctness$. Thus, for a skill, the representation for its prerequisites is calculated as:

$$Pr_i = \frac{\sum_{j=1}^{n} w_j P_j}{\sum_{j=1}^{n} w_j}$$

- n: Number of prerequisites.
- P_j: A student's performance on the jth prerequisite.
- $w_j = \frac{1}{S_j}$: The weight assigned into the jth prerequisite. $S_j$is the correctness of this prerequisite.

Suppose we also have the skill s1's prerequisites are s2 and s3, then using the data from Table 1 the student st1's prerequisite performance on skill s1 is:

$$\frac{0.47 * \frac{1}{1} + 0.3 * \frac{1}{0.5}}{\frac{1}{1} + \frac{1}{0.5}} = 0.36$$

Respectively, the student st2's prerequisite representation value for s1 is 1.

3.3 Defining General Learning Ability

Our approach is to construct a variable, which we refer to as General Learning Ability (GLA), that encapsulates some of the constructs like diligence, home support, raw ability, and so on. GLA refers to a student's latent ability that affects his ability to learn new skill, similar in spirit to the unidimensional trait in Item Response Theory (IRT) [14]. In IRT, a student's trait is assumed measurable; it is measured through a series of adaptive questions given by a tutoring system.

To simplify our work, we measure student's general learning ability as following steps:

a) For each student-skill pair, randomly select the other two started skills. Here a started skill means the student has practiced at least one problem on it;
b) Compute the performance values for the two skills, as described in Section 3.1;
c) Take the average of those two performance values as the general learning ability for this student-skill pair.

Our intuition in defining GLA in this manner is that if the reason for WH's strong gradient with wheel spinning (Figure 3) is due to the knowledge of the prerequisite being important, we would expect GLA to perform poorly. However, if the power of WH comes not from estimating a particular aspect of student knowledge, but rather than providing a proxy measurement for a student's general ability and willingness to learn, we would expect estimating the student's knowledge of two random skills would work as well. We chose to use two random skills since that was the average number of prerequisites, and we wanted to avoid issues with one measure having lower variability (and hence higher reliability) simply by being an aggregate of more skills. One potential drawback of our approach is that two skills is a small number, and in some cases will certainly provide an over- or under-estimate of knowledge for a particular student. However, since our sample size is large enough, 48256 student-skill pairs in total, this approach is unlikely to produce skewed results.

4. RESULTS

4.1 Data Set

The data in this work is from ASSISTments. We tracked all ASSISTments students when they used the system to practice Math problems for almost a full year from September 2010 to July 2011. This data set contains 7591 different students, and we randomly select 4976 of the students (about 2/3 of students) to form our training data set, while the other students comprise the testing data. There are 31301 student-skill pairs in the training set and 16955 in the testing set. In this work, we consider students who fail to achieve mastery within 10 practice opportunities for a skill (including indeterminate cases [1]) as wheel spinning, which results in 20.6% instances in the training set as wheel spinning and 19.2% in the testing set.

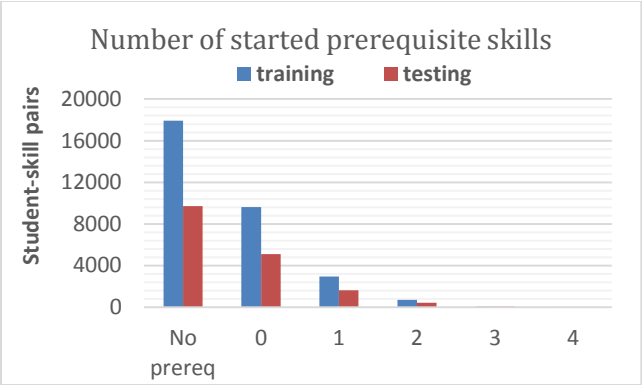


Figure 1. Distribution of number of started prerequisite skills in training set and testing set.

In the training data, there are 177713 problems solved by the students, while 97768 problems in testing data. These problems cover 128 different skills. In the training and testing set, students learn different skills. The maximum number of learned skills by a student is 61, and the average is 6.4. As aforementioned, the prerequisite-to-post skill structure is defined by domain expert as a recommended sequence of topics for instructors. Among the skills in our data set, 66 skills have at least one prerequisite. Some skills have multiple prerequisites, the max number of prerequisites is 8, and the average is 2.4.

However, it is the teacher's choice which skills and in which order to assign to students. Consequently, the majority of student-skill pairs do not have any started prerequisite skills in our data set, as shown in Figure 1. Apparently (and understandably), teachers are less likely to assign review material than to focus on new topics. The maximum number of started prerequisites is 4, and the average is only 0.37. Thus, our experiments will run over three different data sets:

- D1: the whole data set, as depicted in Figure 1, which is splitted into training and testing set.
- D2: the prerequisite data set. This data set excludes the skills that have no prerequisite skills, as identified by the domain expert, from D1. Thus, it is comprised of the points on the x-axis in Figure 1 corresponding to 0, 1, 2, 3 and 4. It is also splitted into training and testing set, and its training set is constructed from the training set in D1 by removing the non-prerequisite skills, while its testing set from testing set in D1 respectively.
- D3: the *started* prerequisite data set, and includes only student-skill pairs where the student has at least begun one of the prerequisites. This data set excludes the skills that have no started prerequisite skills from D2. Thus, it is comprised of the points on the x-axis in Figure 1 corresponding to 1, 2, 3 and 4. Similarly, its training (testing) set is generated from training (testing) set in D2 by removing non-started-prerequsite skills.

The reason for these three datasets is that they answer different research questions. D1 enables us to investigate the impact of prerequisite performance on wheel spinning in an already-existing system in a real-world deployment. That is, how much benefit would we see in the current usage context of the tutor. Unfortunately, that real-world deployment involves teachers assigning no work on most prerequisites, and thus no information about student prerequisite knowledge is available to the model. D2 enables us to examine where there is at least potential benefit. D3 enables us to answer questions about whether a system that had fuller information about prerequisite would perform better at detecting wheel spinning. D3 lets us consider possible changes to policy where teachers are more willing to assign review work, or a system is better able to access past student performance to assess prior knowledge.

4.2 Prerequisite Effect on Wheel Spinning

4.2.1 The Gradient of the Wheel Spinning Ratio

In order to determine how likely a student will be to wheel spin on a skill based on his corresponding prerequisite performance value, we focus on the training set of D3. We separate D3 into 5 bins according to the prerequisite performance value, calculated by the method weighted by hardness. The wheel spinning ratio in each bin is shown in Figure 2, named WS Ratio - WH.

As observed in the figure, there is a strong gradient with respect to the prerequisite performance: students in the bottom 20% of pre-required knowledge exhibited wheel spinning behavior 50% of the time, while those in the top 20% of pre-required knowledge exhibited wheel spinning behavior only 10% of the time. This expresses strong evidence supporting our hypothesis that student's wheel spinning on post-skill results from poor preparation for future learning in terms of prerequisite knowledge [15].

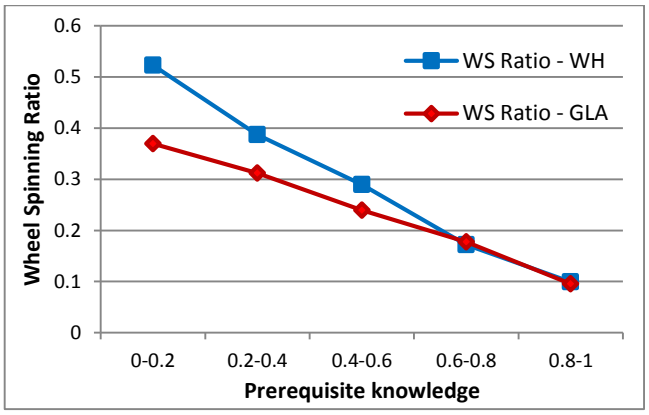


Figure 2. Wheel spinning ratio according with respect to prerequisite knowledge and general learning ability on D3.

4.2.2 Changes in the Model

To test the impact of prerequisite features, we integrated them into the wheel-spinning model described previously. We compare the effects of different factors in the wheel spinning model, Weakest Link (WL), Weighted by Hardness (WH), and General Learning Ability (GLA). Table 3 shows the results of training each model on the training test, and evaluating it on the test set.

In this experiment, we use the Cox and Snell R square [15] and AUC (area under curve) to measure model fit. As we can see, the model does not appreciably change in the data set D1, due to the fact that the part of the data containing started prerequisite skills is such a small component of the data. In D2 and D3, the model is improved slightly by integrating the prerequisite feature, WH or WL. This result supports that prerequisite performance is useful in determining students' wheel spinning status in postrequisite-skills. We can also notice that the model with GLA has the similar results with the ones with WH and WL.

Futhermore, to comare the difference between models, a paired t-test is applied on the results at the student's level of each pair of models, as shown in Table 4. The result shows that adding a

prerequisite factor – WH or WL – into the baseline model makes it performing significantly differently in all data sets, D1, D2, and D3. On the other hand, the model "Baseline+WH" and "Baseline+WL" have the similar results in those three data sets, which also implies these two prerequisite features have similar effect in the wheel spinning model. More interesting, the p-values indicate that the model with GLA is significantly different from the model with WH (or WL respectively) in D1 and D3, but not in D2, and significantly different from the Baseline model in D2, but not in D1 and D3.

Table 3. Measurements of different models.

Model	R Square			AUC		
	D1	D2	D3	D1	D2	D3
Baseline	0.285	0.301	0.264	0.879	0.888	0.884
Baseline +WL	0.285	0.302	0.268	0.879	0.889	0.887
Baseline +WH	0.285	0.302	0.268	0.879	0.889	0.888
Baseline +GLA	0.291	0.306	0.268	0.883	0.891	0.887

Table 4. P-values of paired t-test. In each data set (D1, D2, and D3), we first compute the RMSE for each model predicting over each student. And then the t-test is applied on the RMSE results at the student's level for each pair of models. The p-values in this table are shown in the order (D1, D2, D3).

	Baseline	Baseline+WL	Baseline+WH
Baseline +WL	<0.01,<0.01, <0.01		
Baseline +WH	<0.01,<0.01, <0.01	0.62, 0.1, 0.27	
Baseline +GLA	<0.01,<0.01, 0.21	<0.01,0.29, <0.01	<0.01,0.3, <0.01

4.2.3 Impact of Prerequisite Effect on the Predictive Model

We now move to determining the impact of the prerequisite feature on the predictive model. In our intuition, the prerequisite factor might have strong effect in predicting wheel spinning when a student just starts learning a post-skill, and the effect weakens with time as the student solves problems on the postrequisite skill

In the logistic regression algorithm, researchers typically use the odds ratio, exponential the coefficient, to represent effect of the corresponding feature [15]. Then the coefficient could be also used to represent the effect on the model. Therefore, in this work, we use the coefficient of prerequisite feature to reflect its effect in predicting students' wheel spinning on post-skill.

In this experiment, we group the D3 of training set by amount of practice on the skill, and construct a wheel spinning model for each group. The coefficients of prerequisite feature (for the WH model) in the corresponding models are shown in Figure 3. As we can see, the coefficient representing the impact of prerequisite knowledge has the highest value at the beginning, and it decreases in influence as students obtain more practice on the skill. This result support our intuition that the prerequisite factor is a good predictor for wheel spinning only at the beginning stage of learning post-skill. Thus, prerequisite knowledge is useful for overcoming the cold start problem in student modeling. When a student first starts working on a skill, his performance on that skill provides little basis with whether to classify him as likely to wheel spin or not. In this situation, knowing how he performed on the prerequisite skills provides some information in his ability to master the current material. As the system observes more and more performances on the skill, those performance provide a much more pertinent source of information about the student's likely trajectory, and the relative importance of prerequisite skills diminishes.

The decrease in in predictive performance for the WH coefficient is monotonic and roughly linear. From a standpoint of statistical significance, the WH coefficient is reliably different than 0 for practice opportunities 1 through 7 ($p=0.026$ at the 7^{th} opportunity). At the 8^{th} opportunity, the impact of the WH coefficient has $p=0.51$.

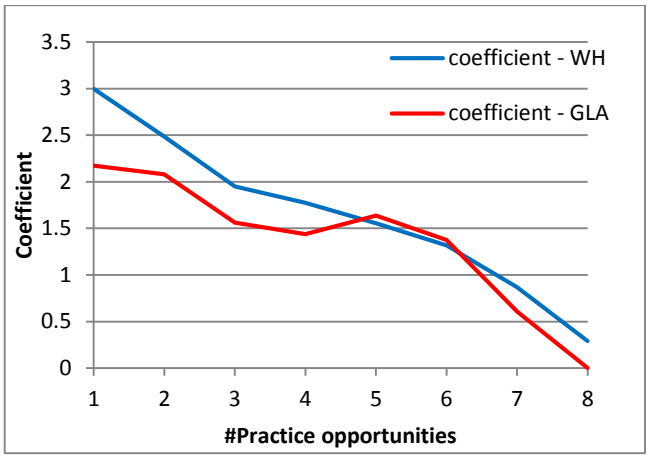


Figure 3. The changes of coefficient with respect to number of practice opportunities on D3.

4.3 Understanding What Prerequisite Performance Really Represents

The performance of the WH feature raises an interesting question: to what does it owe its predictive power. Although we refer to this feature as representing student's prerequisite knowledge, it captures much more than just knowledge. For example, if one student demonstrates strong performance on prerequisite skills and the other does not, those students probably differ in many dimensions beyond knowledge of the skill: diligence in doing math homework, support at home, raw ability at learning new concepts, and perseverance when stuck. Wrapping this bundle of constructs together and calling it "prerequisite knowledge" certainly simplifies discussion, but does a disservice to accuracy. Therefore, we perform a baseline experiment to investigate what prerequisite knowledge represents.

4.3.1 Compare GLA with WH

Since the effects of two prerequisite features, WL and WH, are pretty much the same in the wheel spinning model. Therefore, we will compare only the WH with the GLA. These two features are compared though three different experiments.

The first experiment is to construct wheel spinning ratio gradient for GLA. As we can see in Figure 2, there is the same broad trend for both GLA and WH. For both measures, students with lower general learning ability are more likely to be wheel spinning, which is in accord with our common sense. By comparing the two wheel spinning ratio gradients, we notice that the ratio is the same when the WH and GLA values are high; that is, if a student's performance

is relative high (> 0.6) for WH and GLA, then there is a similar chance the student will wheel spin. However, in the lower range of 0 to 0.6, students are more likely to be wheel spinning according to WH value than the students having the same GLA value. This result suggests that prerequisite factor has stronger correlation with wheel spinning than general learning ability, although general learning ability has strong overlap.

The second experiment is to add the GLA into wheel spinning model and compare the model measurements. According to the results in Table 3, adding the GLA into the baseline model makes more improvement than adding the WH on the data set D1 and D2. This is because the student-skill pairs with pre-required knowledge are very rare in those data sets, while every student-skill pair is assigned with a computed GLA value based on that student's performance on a pair of random skills. The model with GLA and the model with WH on the data set D3 have nearly identical performance.

The third experiment is to compare the effect in the learning procedure. As seen in Figure 3, the GLA coefficient also decreases with respect to the number of practice. But in the first 5 practices, the slope of GLA coefficient is more moderate than the slope of WH coefficient, which defends the statement that the prerequisite factor is useful in predicting wheel spinning at early learning stage. By examine the GLA coefficient Wald statistic p-value, it is also statistically reliable ($p<0.05$) before the 7th practice.

5. DISCUSSION AND FUTURE WORK

It should be noticed that even though we found that prerequisite knowledge is related to wheel spinning on post-skills, the general learning ability also has the similar relation. Therefore, it is hard to identify which factor has a stronger connection with wheel spinning in this data set. This is because of two possible reasons: improper prerequisite structure and indirect prerequisite-post relation.

5.1 Prerequisite Structure

As aforementioned, the prerequisite structure used in this work is defined by domain experts. Through this structure, the experts suggest a general curriculum over all grades, not specified in a single year or a single class. It is certainly possible that our structure is in error either by missing some links and incorrectly creating others. Such errors would impact the results.

Moreover, in the method of computing prerequisite performance for a post-skill, we assume that the prerequisite skill with the worst performance (or the hardest prerequisite skill) has the strongest influence in learning post-skill. However, this assumption might be inappropriate here. Botelho [5] et al. also illustrate in their experiments that the prerequisite relation in some post-skills are not as stable as expected by domain experts.

Therefore, there are two possible ways of improving our experiments. The first one is to construct a prerequisite structure specifically for the data. Previous works have been focused on this area. For example, Vuong et al. [8] introduce a method for finding prerequisite structure within a curriculum. Their method calculates the overall graduation rate for each unit, and regards Unit A as prerequisite knowledge for Unit B if the experience in Unit A promotes graduation rate in Unit B.

The other possible way is to measure the correlation between each prerequisite skill and a post-skill, and then we can obtain which prerequisite skill is most effective in affecting learning post-skill. Vuong et al. also distinguish the prerequisite relationship between significant and non-significant in their work [8].

5.2 Prerequisite-post Relation

Obviously, students' general learning ability influences their performance in both prerequisites and post-skills. Therefore, one might argue that there is no direct causal prerequisite-post relationship. The student who is wheel spun on learning post-skill as well as lack of pre-required knowledge is mainly because he/she has weak learning ability, as shown in Figure 4. In this view, GLA is the primary driver of both prerequisite and postrequisite performance.

According to this argument, a consequent case would be: a student who is wheel spun on a skill, he/she will be wheel spun on every skill, due to the weak learning ability. However, in our data set, the wheel spinning ratio of the students who have at least one wheel spinning case is about 23%. Thus, the GLA is an effective factor in wheel spinning, but not a unique or crucial one. Another drawback of this model is that, for low levels of performance, prerequisite knowledge is more strongly related to wheel spinning than GLA. Therefore, even if GLA is the primary driver, there is apparently some impact of prerequisite knowledge on postrequisite performance, represented by the dotted line in Figure 4.

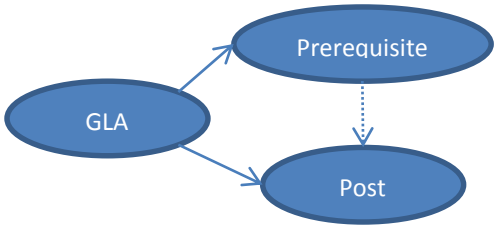


Figure 4. A structure to explain indirect prerequisite-post relationship.

In order to validate the structure in Figure 4, a subtler model should be constructed, in which students' GLA is finely measured. A proper way is to utilize the IRT model to estimate a student's trait; this trait is regarded as the GLA value. And then it is used in predicting if the student will be wheel spinning or not. Meanwhile this trait is updated for each item practiced or for each skill learned. The similar work is in [16], the authors integrate temporal IRT into Knowledge Tracing model, in order to track students' knowledge stage and predict next problem correctness.

6. CONTRIBUTIONS AND CONCLUSION

This work makes two contributions. First, it examines the relationship between prerequisite performance and wheel spinning. One plausible hypothesis for why some students are stuck in the mastery learning cycle is due to inadequate preparation in the building block skills. We found such an association, with students who performed less well on the prerequisite skills being more likely to wheel spin. This work represents an advance over what is known about wheel spinning [1][2].

The second contribution of this work is unpacking what is meant by knowledge of prerequisite skills, and discovering that it is not always related to relevant knowledge. Specifically, by showing that two random skills work approximately as well as prerequisite performance, we show that, for this study, the impact is largely due to general properties of the student than the student's knowledge about particular skills. This reasoning is more than a semantic game, as it directly impacts the conclusions we can draw from our data.

Given just the WH line in Figure 2, a reasonable interpretation is that we can reduce wheel spinning by increasing student

prerequisite knowledge, and we could imagine interventions designed to target such. Given the additional context of the results for GLA, we realize that most of the effect attributed to prior knowledge is really just how well the student learns math in general. Unfortunately, interventions to target diligence, grit, math ability, and home support are outside the scope of plausible interventions to deliver with an ITS. However, the difference in the gradients of the two lines suggests there is some benefit from improving student knowledge to at least a moderate level to reduce wheel spinning. This analysis also raises the question of how much work reporting effects related to student prior knowledge is really talking about some other construct than knowledge. Unless the difference in knowledge is caused by a randomized manipulation, differences in knowledge are a proxy for a collection of variables. Hopefully this work will spur EDM researchers to more carefully investigate the meaning of the constructs they are reporting.

In conclusion, this paper investigates the effect of prerequisite performance on wheel spinning and finds that they are related. The addition of prerequisite or GLA features provides a small enhancement in predictive accuracy to our wheel spinning model, improving R2, on skills for which we have prerequisite data, from 0.264 to 0.268, and AUC from 0.884 to 0.888. The baseline model results are quite strong for ITS research, so third-decimal improvement in both metrics is fairly good.

This work also found that prerequisite performance and GLA are both effective for overcoming the cold start problem in student modeling. When students begin working on a skill, the tutor has little knowledge of the student's capabilities on that skill. We found that the new factors in our model had the greatest impact when students were first starting to work with a skill, and diminish in importance as we acquire additional data about his knowledge of the skill.

7. ACKNOWLEDGEMENTS

We acknowledge funding from NSF (# 1440753, 1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024) grant for ASSISTments and support of the author.

REFERENCES

[1] J. E. Beck and Y. Gong, "Wheel-Spinning:Students Who Fail to Master a Skill," in *Proceedings of 16th International Conference, AIED 2013*, Memphis, TN, USA, 2013.

[2] J. E. Beck and M. M. T. Rodrigo, "Understanding Wheel Spinning in the Context of Affective Factors," in *Proceedings of 12th International Conference, ITS 2014*, Honolulu, HI, USA, 2014.

[3] Y. Gong and J. Beck, "Towards Detecting Wheel-Spinning: Future Failure in Mastery Learning," in *Learning at Scale 2015*, 2015.

[4] C. Carmona, E. Millán, J. L. Pérez-de-la-Cruz, M. Trella and R. Conejo, "Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model," in *Proceedings of 10th International Conference, UM 2005*, Edinburgh, Scotland, UK, 2005.

[5] A. Botelho, H. Wan and N. Heffernan, "The Prediction of Student First Response Using Prerequisite Skills," in *Learning at Scale 2015*, 2015.

[6] E. Brunskill, "Estimating Prerequisite Structure From Noisy Data," in *EDM*, 2011.

[7] P. I. Pavlik Jr, H. Cen, L. Wu and K. R. Koedinger, "Using Item-Type Performance Covariance to Improve the Skill Model of an Existing Tutor," in *Proceedings of the 1st International Conference on Educational Data Mining*, Montreal, Canada, 2008.

[8] A. Vuong, T. Nixon and B. Towle, "A Method for Finding Prerequisites Within a Curriculum," in *EDM*, 2011.

[9] P. I. Pavlik Jr, H. Cen, L. Wu and K. R. Koedinger, "Performance Factors Analysis - A New Alternative to Knowledge Tracing," in *the 14th International Conference on Artificial Intelligence in Education*, Brighton, UK, 2009.

[10] I. Arroyol and B. P. Woolf, "Inferring learning and attitudes from a Bayesian Network of log file data," in *Artificial Intelligence in Education*, 2005.

[11] Y. Gong, J. E. Beck, N. T. Heffernan and E. Forbes-Summers, "The impact of gaming (?) on learning at the fine-grained level," in *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010)* , 2010.

[12] R. S. J. d. Baker, A. T. Corbett, I. Roll and K. R. Koedinger, "Developing a generalizable detector of when students game the system," *User Modeling and User-Adapted Interaction*, vol. 18, no. 3, pp. 287-314, 2008.

[13] A. T. Corbett and J. R. Anderson, "Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge," *User Modeling and User-Adapted Interaction*, pp. 253-278, 1995.

[14] S. E. Embretson and S. P. Reise, Item Response Theory for Psychologists, Psychology Press, 2013.

[15] R. S. J. D. Baker, S. M. Gowda, A. T. Corbett and J. Ocumpaugh, "Towards automatically detecting whether student learning is shallow," in *Intelligent Tutoring Systems*, 2012.

[16] D. W. Hosmer Jr and S. Lemeshow, Applied Logistic Regression, John Wiley & Sons, 2004.

[17] Y. Huang, J. González-Brenes and P. Brusilovsky, "General Features in Knowledge Tracing to Model Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge," in *Educational Data Mining 2014*, 2014.

Comparing Novice and Experienced Students within Virtual Performance Assessments

Yang Jiang, Luc Paquette, Ryan S. Baker
Teachers College, Columbia University
525 W 120th Street
New York, NY 10027
yj2211@tc.columbia.edu
paquette@tc.columbia.edu
baker2@exchange.tc.columbia.edu

Jody Clarke-Midura
Utah State University
2830 Old Main Hill
Logan, UT 84322
jody.clarke@usu.edu

ABSTRACT

Inquiry skills are an important part of science education standards. There has been particular interest in verifying that these skills can transfer across domains and instructional contexts [4,15,16]. In this paper, we study transfer of inquiry skills, and the effects of prior practice of inquiry skills, using data from over 2000 middle school students using an open-ended immersive virtual environment called Virtual Performance Assessments (VPAs) that aims to assess science inquiry skills in multiple virtual scenarios. To this end, we assessed and compared student performance and behavior within VPA between two groups: novice students who had not used VPA previously, and experienced students who had previously completed a different VPA scenario. Our findings suggest that previous experience in a different scenario prepared students to transfer inquiry skills to a new one, leading these experienced students to be more successful at identifying a correct final conclusion to a scientific question, and at designing causal explanations about these conclusions, compared to novice students. On the other hand, a positive effect of novelty was found for motivation. To better understand these results, we examine the differences in student patterns of behavior over time, between novice and experienced students.

Keywords

Virtual environment, science inquiry, educational data mining, sequential pattern mining, transfer, novelty effect.

1. INTRODUCTION

One of the important goals for science education is to help students develop the scientific knowledge and practices needed to actively and effectively engage in science inquiry. As such, science inquiry skills have been a critical component of the K-12 science curriculum standards [18]. It is particularly crucial that students acquire inquiry skills which are not specific to a domain or instructional context, but which can transfer broadly, preparing students for using science and understanding science in their future schooling, and in their lives [4, 15, 16].

With the increasing popularity of online learning systems that engage learners in science inquiry activities [e.g., 7, 21], Educational Data Mining (EDM) techniques have proven effective in automatically assessing science inquiry skills. Sao Pedro et al. demonstrated that science inquiry skills can be assessed within online learning activities using EDM, predicting future performance not only within the same domain [21], but also across domains [22].

Many studies of student inquiry behavior have been conducted within open-ended online learning environments, such as virtual environments, in which users have the freedom to decide their own inquiry behaviors. This, combined with the fact that these open-ended environments are typically more loosely-scaffolded and coarser-grained than more tightly-scaffolded systems such as intelligent tutoring systems or simulations [e.g., 21], makes the assessment of science inquiry in these contexts challenging. Sequential Pattern Mining [1], a methodology that has been extensively used in EDM [23], has shown potential in discovering complicated patterns of learning behavior within open-ended learning environments. For example, Kinnebrew and colleagues [13] applied sequential pattern mining techniques to log data produced by students engaging in activities within Betty's Brain, an open-ended learning environment for science learning. This allowed them to study the differences in students' productive and unproductive learning behaviors by identifying frequent sequential patterns related to the use of concept maps and determining which sequential patterns were characteristic of high-performing students as compared to low-performing students. Differential pattern mining was also used by Sabourin and colleagues [20] to analyze the differences in inquiry behaviors utilized by learners depending on their level of self-regulation within a virtual environment. In another study, Gutierrez-Santos et al. [10] conducted analysis of student actions to detect repetitive sequences in an open-ended learning environment.

Another EDM approach that has proven useful for the study of inquiry behaviors in open-ended contexts involves in-depth analysis of features distilled from log data. For instance, Baker and Clarke-Midura [2] distilled a set of features related to inquiry behavior from log data in Virtual Performance Assessments (VPAs), an open-ended immersive virtual environment used in the current study, to develop predictive models of student success on two inquiry tasks. The current study combines both sequential pattern mining and analysis of features related to science inquiry to study transfer of inquiry skills. In doing so, we also analyze differences in inquiry behavior between novice students and experienced students.

The degree of student experience with an environment can also be hypothesized to have important impact on their inquiry. Clark [6] argued that novelty effect occurs when new computer programs

are introduced. In those cases, the novel computer programs initially attract student attention, leading to increased efforts invested, persistence, motivation, and achievement gains. Previous studies [e.g., 8, 12, 24] indicated that students showed greater initial enthusiasm and motivation in classrooms when novel educational technologies were introduced. This enthusiasm gradually diminished as students were more familiar with the technologies and the initial novelty effect wore off. Therefore, in our study, we investigate whether relative novelty created by the introduction of a new 3D virtual environment will lead to differences in motivation and learning between novice students and experienced students. We also study the relationship between the potential novelty effect and inquiry skills.

To research these questions, we assess and compare student performance and behavior within VPA between two groups: novice students who had not used VPA previously, and more experienced students who had previously spent one class session completing a different VPA scenario. We compare student performance on two inquiry skills – identifying a correct final claim and designing causal explanations. We also compare student responses to a motivation survey between the two groups. Finally, we analyze the difference in student behavior between the two groups using differential sequence mining.

2. VIRTUAL PERFORMANCE ASSESSMENTS

This study was conducted within the context of Virtual Performance Assessments (VPAs; see http://vpa.gse.harvard.edu). VPAs are online 3D immersive virtual environments, designed using the Unity game development engine [26] that assess middle school students' science inquiry skills, in line with state and national standards for science content and inquiry processes. Within VPAs, whose interface is similar to that of video games, students engage in authentic inquiry activities and solve scientific problems by navigating around the virtual environment as an avatar, making observations, interacting with non-player characters (NPCs), gathering data, and conducting laboratory experiments. VPAs enable automated and non-intrusive collection of process data (logged actions and behaviors) and product data (student final claims), facilitating the capture and assessment of science inquiry *in situ*.

Multiple VPA assessment scenarios have been developed. In this study, two scenarios were used, the frog scenario and the bee scenario. In the frog scenario (see Figure 1), students are presented with a six-legged frog in the virtual environment and have to collect and reason through evidence to determine what is causing the frog's mutation, selecting from a set of possible causal factors including parasites (the correct causal explanation), pesticides, pollution, genetic mutation, and space aliens. In this scenario, students can talk with NPCs from four virtual farms who provide conflicting opinions, collect items such as frogs, tadpoles, and water samples at each farm, run laboratory experiments on water quality, frog blood and DNA, and read informational pages from a research kiosk. Once students think that they have sufficient data, they submit a final conclusion on the causal factor resulting in the mutation, and justify their final claim with supporting evidence. In the bee scenario, students must determine what causes the death of a local bee population. Similar to the frog scenario, they can talk with NPCs from four different farms, read informational pages at the research kiosk, and conduct tests (e.g., nectar test, protein test, genetic test) on the items they have collected at the farms (e.g., nectar samples, bees, larvae). By the end of the assessment, students choose a final claim about the cause of the bee deaths from possible hypotheses including genetic mutation (the correct causal factor), parasites, pesticides, pollution, and space aliens, and support their final claim with evidence. The activities in each VPA scenario are deliberately similar, allowing researchers to assess performance of the same inquiry practices in different contexts.

Figure 1. Screenshots of the VPA frog scenario.

3. DATA SET

Data for this study was composed of action logs produced by middle school students who used Virtual Performance Assessments within their science classes at the end of the 2011-2012 school year. A total of 2,431 students in grades 7-8 (12-14 years old) from 138 science classrooms (40 teachers) participated in this study. These students were from a diverse range of school districts in the Northeastern and Midwestern United States, and Western Canada. A total of 1,985 students completed the frog scenario and 2,023 students completed the bee scenario, with 1,579 students completing both scenarios. Overall, students completed 423,616 actions within the frog scenario and 396,863 actions within the bee scenario. They spent an average of 30 minutes and 47 seconds (*SD* = 14 minutes, 6 seconds) in the frog scenario and an average of 26 minutes and 5 seconds (*SD* = 12 minutes, 27 seconds) in the bee scenario.

The 2,431 students were randomly assigned to begin with either the frog scenario or the bee scenario. Two weeks later, they were assigned to complete the other scenario. Therefore, within each scenario, participants could be put into two groups – novice users who were using VPA for the first time (*novice* group), and experienced users who had previously experienced the other VPA scenario (*experienced* group). Accordingly, among the 1,985 students who completed the frog scenario, 1,232 completed the frog scenario as their first scenario (frog-novice) and 753 had previous experience in the bee scenario (frog-experienced). Among the students who completed the bee scenario, 1,198 students had no previous experience in the frog scenario (bee-novice), whereas 825 had previous experience in the frog scenario (bee-experienced). Student actions and performance in the virtual environment were logged as they worked within each VPA scenario and used for later analyses.

4. OVERALL ANALYSIS

In this section, we compare student performance on identifying a correct final claim and constructing causal explanations, the amount of time spent on VPA, and students' motivation level, between the novice group and the experienced group, within each VPA scenario.

4.1 CFC and DCE Performance

To explore the transfer of student science inquiry skills between scenarios, two measures of student performance within the VPAs were collected and compared between the two groups of students within each scenario: 1) the correctness of the student's final claim (CFC) on the cause of the six-legged frog or the death of the

bees; and 2) student's success in designing causal explanations (DCE) for why that claim is correct.

In each VPA, students submitted a final claim by choosing from five possible causal factors. A student's final claim was considered correct if the student concluded that the mutation of the six-legged frog was caused by parasites (correct causal factor), or that the bee deaths were caused by genetic mutation (correct choice). Otherwise, if the student selected the other potential hypotheses, the student's final claim was considered incorrect. Overall, 29.6% of students correctly concluded that parasites led the frog to have six legs, and 28.3% of students made a correct claim on what was killing the bee population. In this paper, a chi-square test was conducted to compare student CFC performance between the two groups in each scenario.

In the bee scenario, 34.8% of experienced students who had previously used the frog scenario identified correctly that genetic mutation was killing the bees, while 23.9% of novice students (*without* prior experience in the frog scenario) made the correct final conclusion. This difference was statistically significant according to a chi-square test, $\chi^2(1, N = 2023) = 28.67$, $p < .001$. Logistic regression results revealed that the odds of making a correct final claim for experienced students (0.533) was statistically significantly larger than the odds for novice students (0.314) by 70%. This suggested that the students transferred what they learned about how to make a correct final claim from the frog scenario to the bee scenario.

Similarly, in the frog scenario, a statistically significantly higher percentage of experienced students (33.2%) made a correct final claim than the percentage of novice students (27.5%) who made a correct conclusion, $\chi^2(1, N = 1985) = 7.45$, $p = .006$. Logistic regression results indicated that previous experience in the bee scenario significantly improved the odds of making a correct final claim in the frog scenario by 31.5% (odds = 0.378 for novice group and 0.497 for experienced group).

The DCE measure evaluates student ability in supporting final conclusions with evidence. By the end of the assessment in each scenario, students needed to select the evidence that supported their claims from the data they had collected within the virtual world and the results of laboratory tests they had conducted. They were then presented with all possible data (including data that the students did not collect/conduct) and asked to identify the evidence supporting their claim. In each VPA scenario, most evidence was consistent with the correct causal claim. However, for the incorrect claims, there was often evidence consistent with those claims along with counter-evidence that conclusively disproved those hypotheses. Therefore, even if students were unsuccessful in identifying the correct final conclusion, partial credit would be awarded to them for the quality and quantity of the causal evidence they identified in support of their claim from the non-causal data and results. Student success in selecting evidence and constructing causal explanations were aggregated into a single composite DCE measure that ranges from 0 to 100%, by averaging across the use of each piece of evidence. The mean DCE score for the frog scenario was 50.0% (*SD* = 23.3%), and the average DCE score for the bee scenario was 46.1% (*SD* = 21.4%). A two-tailed Mann-Whitney U test, a nonparametric alternative to t-test, was then conducted to compare student ability in designing causal explanations between the two groups in each scenario.

Results of the Mann-Whitney U test comparing the DCE score between the two groups in the bee scenario suggested that the experienced group had a significantly higher average DCE score (*M* = 48.9%, *SD* = 19.3%) than the novice group (*M* = 44.2%, *SD* = 23.8%), *U* = 453873, *Z* = -3.12, *p* = .002. Further analyses revealed that the difference in DCE performance was dependent on the correctness of final claims. Among students who made a correct final claim in the bee scenario, the experienced group achieved significantly higher DCE scores (*M* = 75.1%, *SD* = 18.3%) than the novice group (*M* = 68.1%, *SD* = 20.5%), *U* = 32448.5, *Z* = -4.34, *p* < .001. However, among students who did not make a correct final claim, the novice group showed higher DCE scores (*M* = 36.7%, *SD* = 11.2%) than the experienced group (*M* = 34.9%, *SD* = 11.4%), *U* = 223797, *Z* = -2.80, *p* = .005.

In the frog scenario, student performance in designing causal explanations for the novice group (*M* = 49.7%, *SD* = 22.7%) was not statistically significantly different from the experienced group (*M* = 50.6%, *SD* = 24.3%), *U* = 454398, *Z* = -.76, *p* = .446.

4.2 Time

As each VPA scenario logged the timing of each student starting and exiting the virtual environment, we also compared the total amount of time students spent within VPA recorded by the log data between the novice group and the experienced group, by employing one-way ANOVA.

An analysis of variance showed that, on average, novice students without previous experience in the frog scenario spent significantly more time in the bee scenario (*M* = 27 minutes, 43 seconds, *SD* = 11 minutes, 56 seconds) than experienced students who had used the frog scenario (*M* = 23 minutes, 43 seconds, *SD* = 12 minutes, 48 seconds), *F* (1, 2021) = 51.64, *p* < .001. On the other hand, the total amount of time spent in the frog scenario by novice students (*M* = 30 minutes, 56 seconds, *SD* = 14 minutes, 24 seconds) and experienced students (*M* = 30 minutes, 33 seconds, *SD* = 13 minutes, 35 seconds) was not statistically significantly different (*F* (1, 1983) = .36, *p* = .548).

4.3 Motivation

In this study, students completed an online motivation survey shortly after they finished the VPA assessment for each scenario. Student responses to the survey were analyzed to better understand the impact of experience with the environment on learning and motivation. The survey was adapted from the Intrinsic Motivation Inventory [IMI; 27] and the Player Experience of Need Satisfaction [PENS; 19] survey and was comprised of 27 six-point Likert-type items that aimed to measure seven components related to student motivation, autonomy, and in-game immersion: interest/enjoyment, perceived competence, effort/importance, pressure/tension, value/usefulness, presence/immersion, and autonomy. Items were slightly modified to fit the specific activity in this game-like environment. Student subscale scores were calculated by averaging across all items on each subscale. One-way ANOVA was applied to assess whether there were any systematic differences in student motivation between the novice group and the experienced group within each VPA scenario. Given the substantial number of statistical tests, we controlled for the proportion of false positives by applying Storey's q-value method [25] (calculated using the QVALUE package for R).

Analyses of motivational survey results (see Table 1) indicated that, on average, novice students scored significantly higher on the interest/enjoyment subscale than experienced students in both scenarios (*F*(1, 1800) = 50.02, *q* < .001 for the frog scenario; *F*(1, 1740) = 27.67, *q* < .001 for the bee scenario). Similarly, students

in the novice group had a significantly higher level of perceived effort invested to the VPA activity and perceived importance of the activity than students in the experienced group ($F(1, 1800) = 25.41$, $q < .001$ for the frog scenario; $F(1, 1740) = 18.94$, $q < .001$ for the bee scenario). Novice students also regarded the VPA activity as more useful and valuable than experienced students, $F(1, 1800) = 19.37$, $q < .001$ for the frog scenario; $F(1, 1740) = 4.66$, $q = .019$ for the bee scenario. Finally, novice students also had significantly higher presence/immersion, autonomy, and tension/pressure subscale scores than the experienced students, indicating that they were more immersed in the virtual environment, and felt a higher sense of autonomy and a higher level of tension/pressure than experienced students. These corresponded to previous findings on novelty effect [8, 12].

Table 1. Average subscale scores on the motivational survey (standard deviations in parentheses) by condition. Differences that are sig. after post-hoc controls ($q < 0.05$) are marked by *.

Subscale	Frog-N	Frog-E	*F (q)*	Bee-N	Bee-E	*F (q)*
int/enj	4.47 (1.32)	3.98 (1.55)	50.02* (<.001)	4.26 (1.42)	3.87 (1.56)	27.67* (<.001)
comp	4.28 (1.21)	4.23 (1.37)	0.73 (.213)	4.13 (1.27)	4.14 (1.37)	0.006 (.473)
eff/imp	4.38 (1.19)	4.06 (1.44)	25.41* (<.001)	4.21 (1.30)	3.91 (1.49)	18.94* (<.001)
val/use	4.07 (1.41)	3.74 (1.62)	19.37* (<.001)	3.84 (1.51)	3.67 (1.64)	4.66* (.019)
pres/ten	1.86 (1.25)	1.72 (1.39)	4.62* (.019)	1.85 (1.29)	1.69 (1.38)	5.86* (.011)
pres/imm	3.51 (1.36)	3.16 (1.53)	24.72* (<.001)	3.36 (1.42)	3.13 (1.53)	10.14* (.001)
auto	4.26 (1.29)	3.82 (1.55)	41.12* (<.001)	4.01 (1.41)	3.76 (1.56)	11.42* (.001)

Note. Frog-N = frog-novice, Frog-E = frog-experienced, Bee-N = bee-novice, Bee-E = bee-experienced. Int/enj=interest/enjoyment, comp=perceived competence, eff/imp=effort/importance, pres/ten=pressure/tension, val/use=value/usefulness, pres/imm=presence/immersion, auto= autonomy.

5. USAGE ANALYSIS

In the previous section, differences were found in motivation and learning outcomes between novice and experienced students. In the current section, we aim to go beyond just looking at whether previous experience in VPA improved student inquiry performance, and instead look into whether more experienced students used VPAs differently than less experienced students. For example, this will allow us to determine whether the higher success for experienced students within VPAs was related to the acquisition and transfer of science inquiry skills, or whether it was merely the result of increased familiarity and proficiency with using the system and tools than novice users.

We studied these questions by investigating the prevalence of specific behaviors between groups, and by applying sequential pattern mining to identify and compare the frequent sequential patterns of student actions between groups.

5.1 Comparing Behaviors Between Groups

In order to understand student behavior, and how it differed between groups, a set of 30 semantically meaningful features of student behavior thought to potentially differ between groups were distilled from raw interaction data and were compared between the novice and experienced groups in each scenario. These features were a subset of the 48 features that were used to build models predicting a student's CFC and DCE performance within the frog scenario in [2]. Examples of these features will be given in the following paragraphs.

After distilling the 30 features from each student's interaction logs, t-tests were conducted to compare the value of each feature between the experienced and novice groups, within each scenario. Storey's q-values [25] were calculated to control for multiple comparisons. Table 2 presents the average values of 10 features that strongly differentiated between groups.

According to the results, features representing the maximum or average fullness of a student's backpack in the frog scenario, both including repeats (e.g. picking up two green frogs counts as two objects), and not including repeats (e.g. two green frogs counts as one object), had significantly higher value for the novice group than the experienced group. Similar results were found in terms of the number of times a student went to the lab to run tests, the number of different (types of) non-sick frogs that the student took to the lab at the same time, the number of times that lab water was taken to the lab, and the percentage of time the student spent at farms to collect evidence in the frog scenario. Similarly, novice students in the bee scenario had higher values on all these features than experienced students. This suggested that novice students collected significantly more data for testing and spent a larger proportion of time on collecting evidence in farms than the experienced students in both scenarios. This finding was consistent with the higher motivation level of novice students (in both scenarios) and the longer time they spent working on VPA

Table 2. Comparisons of features between novice group and experienced group. Sig. differences ($q<0.05$) are marked by *.

Feature	Frog-N	Frog-E	t	q	Bee-N	Bee-E	t	q
The number of times student went to the lab	6.66	5.14	6.81	<.001*	16.37	12.71	8.97	<.001*
Maximum number of items (including repeats) in backpack	7.48	6.69	11.25	<.001*	6.03	4.76	11.57	<.001*
Maximum number of items (not including repeats) in backpack	7.45	6.65	11.68	<.001*	8.54	7.28	12.27	<.001*
Average number of items (including repeats) in backpack	4.77	4.02	11.39	<.001*	3.86	3.06	11.91	<.001*
Average number of items (not including repeats) in backpack	4.75	4.00	11.50	<.001*	6.17	5.14	11.61	<.001*
Number of times that lab water/nectar was taken to the lab	0.42	0.38	2.11	.022*	1.69	0.93	8.31	<.001*
Number of different (types of) non-sick frogs/bees student took to the lab at the same time	1.87	1.70	2.34	.014*	4.32	3.90	4.09	<.001*
How long, on average, did students spend reading information pages? (average per read)	15.28	17.17	-0.72	.146	11.93	13.93	-2.07	.027*
How long, in total, did student spend reading information page on correct hypothesis?	32.33	35.13	-0.70	.146	23.45	27.46	-2.20	.021*
Percentage of time student spent at farms	0.29	0.26	4.43	<.001*	0.34	0.31	5.46	<.001*

(in the bee scenario).

Despite the fact that the novices collected more data and spent more total time within the VPA bee scenario, they spent significantly less time on reading an information page at the research kiosk each time they accessed the page (M = 11.93 seconds, SD = 17.69 seconds) than experienced students (M = 13.93 seconds, SD = 25.48 seconds), t (2021) = -2.07, q = 0.027, *Cohen's D* = 0.15. In specific, experienced students spent more time in total reading the information page on the correct hypothesis – genetic mutation (M = 27.46 seconds, SD = 46.51 seconds) compared to novice students (M = 23.45 seconds, SD = 35.46 seconds), t (2021) = -2.20, q = 0.021, *Cohen's D* = 0.11. Gaining more information about the correct hypothesis might have contributed to the students' domain-specific knowledge base, which had been found to be crucial for problem solving and the development of expertise [5]. However, the corresponding pattern was not statistically significant for the frog scenario, probably due to higher standard deviations.

5.2 Sequential Pattern Mining

In this section, we investigate patterns in behavior by the two groups, over time. Prior to performing sequential pattern mining, detailed raw action log data were transformed into more abstract sequences. This involved three steps. First, a set of actions related to science inquiry were identified from the log files, including picking up and inspecting objects (e.g., frogs, tadpoles, bees, larvae, water sample, nectar sample) within VPA (*inspect*), talking with NPCs (*talk*), saving objects to backpack (*save*), discarding objects (*discard*), opening and reading informational pages at the research kiosks (*read*), running laboratory tests (*blood/protein test, water/nectar sample test, genetic test*), reviewing and looking at test results (*look*), starting to answer final questions (*start final questions*), and submitting a final claim (*final claim*). Some actions that were irrelevant to the inquiry process, such as selecting an avatar, closing the scratchpad, and entering/exiting a specific area were filtered out from the raw interaction data. Second, as in [13], repeated actions that occurred more than once in succession were distinguished from a single action and were labeled as the "action" followed by the "-MULT" suffix. This adjustment prevents frequent patterns from being overlooked merely due to differences in how many times the action is repeated. Last, the actions were represented as sequences of actions for each student in each group.

Simple two-action sequential patterns were identified using the arules package [11] within the statistical software program R. Arules was used to determine the most frequent short sequences of two actions by selecting the temporal associations of one specific action and a subsequent action with the highest support and confidence. In this study, sequential patterns of consecutive actions were selected with the cut-off thresholds of support = 0.0005 and confidence = 0.1.

In the frog scenario, a total of 51 short sequential patterns (length = 2) were identified that met the minimum support and confidence constraints within the novice group; 54 patterns were identified within the experienced group. In the bee scenario, 55 short sequential patterns met the minimum constraints within the novice group; 59 were selected within the experienced group. These patterns were similar across the 4 conditions, and most had support and confidence considerably higher than the threshold. They were then ordered according to their *Jaccard* similarity coefficient – a measure of the patterns' interestingness [17] that was found to be the most highly correlated with human judgments [3] – to find interesting sequential patterns. According to [3], lower *Jaccard* measures indicated higher interestingness.

To facilitate the comparison of the frequency measures between the novice group and the experienced group, the support and confidence for each pattern were calculated separately for each student. Mann-Whitney U tests that controlled for multiple comparisons were then conducted to compare the metric values between two groups in each scenario.

Table 3 presents the comparison of the support and confidence levels of 9 frequent sequential patterns with low *Jaccard* measure (indicating high interestingness) across conditions that were considered as meaningful due to the nature of the actions they contained. The sequential patterns with the lowest *Jaccard* included patterns related to making final claims (*final claim*) or starting to answer final questions (*start final questions*) and reading informational pages (*read*), such as "*final claim → read-MULT*", "*final claim → read*", "*read-MULT → final claim*", "*start final questions → read-MULT*", and "*start final questions → read*". These patterns indicated that students tended to review research and read informational pages as resources to assist with their decision-making before submitting a final claim, or that they used the research information to check and monitor the claims they had just made. All these 5 patterns appeared to have higher support for experienced students than novice students within each

Table 3. Comparison of the support and confidence of 9 frequent sequential patterns between novice and experienced conditions. Average support/confidence values, and post-hoc controlled sig. of tests are presented. Sig. differences (q<0.05) are marked by *.

	support			confidence			support			confidence		
Pattern	**Frog-N**	**Frog-E**	**q**	**Frog-N**	**Frog-E**	**q**	**Bee-N**	**Bee-E**	**q**	**Bee-N**	**Bee-E**	**q**
final claim → read-MULT	.0033	.0043	.420	.296	.313	.594	.0030	.0036	.619	.326	.298	.420
read-MULT → final claim	.0061	.0074	.584	.114	.109	.619	.0055	.0064	.675	.101	.109	.594
final claim → read	.0020	.0026	.675	.164	.158	.675	.0014	.0024	.018*	.142	.193	.107
start final questions→ read-MULT	.0046	.0047	.594	.282	.261	.594	.0044	.0049	.675	.274	.257	.594
start final questions → read	.0029	.0033	.682	.160	.167	.675	.0025	.0027	.675	.147	.142	.675
look-MULT → read-MULT	.0027	.0032	.718	.143	.176	.517	.0028	.0030	.594	.141	.189	.309
look → read	.0025	.0028	.711	.103	.142	.214	.0017	.0021	.675	.080	.107	.361
look → read-MULT	.0027	.0033	.675	.113	.158	.073	.0019	.0027	.594	.105	.155	.018*
look-MULT → read	.0021	.0021	.594	.104	.117	.675	.0021	.0017	.018*	.106	.101	.420

scenario, but most of the differences were not statistically significant. In the bee scenario, the pattern *final claim → read* showed significantly higher support and marginally significantly higher confidence for the experienced group than the novice group (for *support*, $Ms = 0.024$ and 0.014, $U = 474169.5$, $Z = -3.03$, $q = 0.018$; for *confidence*, $Ms = 0.193$ and 0.142, $U = 46833.5$, $Z = -2.32$, $q = 0.107$). This finding indicated that experienced students who had previously used the frog scenario were more likely to review research and read information, possibly to monitor their answers and reflect on previous steps [cf. 15], after submitting a final claim in the bee scenario than novice students. However, this trend was not replicated in the frog scenario (for *support*, $Ms = 0.0026$ and 0.0020, $U = 462294.5$, $Z = -.23$, $q = .675$; for *confidence*, $Ms = 0.158$ and 0.164, $U = 58423.5$, $Z = -.32$, $q = .675$).

Another four interesting sequential patterns corresponded to looking at laboratory test results (once or repeatedly), followed by reading informational pages (once or repeatedly) (i.e., *look-MULT → read-MULT, look → read, look → read-MULT, look-MULT → read*). For three out of the four patterns, both the support and the confidence for the experienced group were higher than those for the novice group in both scenarios. For the pattern *look → read-MULT*, the confidence for the experienced group was statistically significantly higher than that for the novice group in the bee scenario and marginally higher than confidence for the novice group in the frog scenario (in bee scenario, $Ms = 0.105$ and 0.155, $U = 94500.5$, $Z = -3.09$, $q = .018$; in frog scenario, $Ms = 0.113$ and 0.158, $U = 111697.5$, $Z = -2.53$, $q = .073$). That is, experienced students were more likely to read multiple research information pages on possible causal factors immediately after looking at the results of lab tests. This is consistent with results from previous studies on the development of expertise, where experts were found to be more opportunistic in using resources and exploit more available sources of information than novices [9]. The higher relative frequency of reading research information, which might help experienced students interpret laboratory test results and facilitate the acquisition of domain-specific knowledge [4], might have contributed to their higher success on making correct final claims than novice students.

In addition to two-action patterns, a differential sequence mining technique developed by Kinnebrew and colleagues [13] was utilized for identifying longer sequential patterns (length > 2) that occurred with significantly different frequencies between the two groups. This methodology used sequence support (*s-support*) and instance support (*i-support*) as frequency measures. S-support is defined as the percentage of sequences in which the pattern occurs [13]. It is different from the standard metric *support* in that s-support measures the percentage of students whose action sequence contained the specific pattern, regardless of the frequency of occurrence within each sequence for each student. The i-support corresponds to the number of times a given pattern occurs, without overlap, within a student's sequence of actions. A set of most frequent sequential patterns that met the s-support threshold was identified within each group by employing Kinnebrew et al.'s sequential pattern mining algorithm [13]. The i-support value of each pre-identified pattern was then calculated for each sequence in each group, after which t-tests comparing the mean i-support between the groups were conducted and q-value post-hoc control [25] was applied to select significantly differentially frequent patterns.

The 25 most differentially frequent long patterns with at least three consecutive actions were identified in the frog scenario and the 32 differentially frequent long patterns were identified in the bee scenario by employing a cutoff s-support of 50% and a cutoff q-value of 0.05 for comparison of pattern usage between two groups. 14 out of the 25 long patterns in the frog scenario and 16 out of 32 long patterns in the bee scenario were common (i.e., met the 50% s-support threshold) for both groups, with relatively higher usage in the novice group. 11 long patterns in frog scenario and 16 in the bee scenario were frequently used only by students in the novice group. All differentially frequent long patterns had a

Table 4. Top differentially frequent patterns between the novice group (nov) and the experienced group (exp).

Scenario	Pattern	s-support		i-support			Frequent
		nov	exp	nov	exp	q	
Frog	talk-MULT → inspect → save → inspect → save	0.58	0.36	0.78	0.45	<.001	nov
	talk-MULT → inspect → save → inspect	0.59	0.37	0.79	0.46	<.001	nov
	save → discard → inspect → save	0.53	0.36	0.74	0.48	<.001	nov
	inspect → save → discard → inspect	0.53	0.36	0.75	0.49	<.001	nov
	inspect → save → discard → inspect → save	0.53	0.36	0.74	0.48	<.001	nov
	talk-MULT → inspect → save	0.78	0.53	1.25	0.70	<.001	both
	inspect → save → talk	0.78	0.60	1.50	0.99	<.001	both
	discard → inspect → save	0.82	0.62	1.97	1.31	<.001	both
	inspect → save → discard	0.78	0.60	1.74	1.19	<.001	both
	talk → inspect → save	0.78	0.63	1.56	1.10	<.001	both
Bee	talk-MULT → inspect → save → inspect → save → inspect	0.59	0.27	0.72	0.32	<.001	nov
	talk-MULT → inspect → save → inspect → save → inspect → save	0.59	0.27	0.71	0.32	<.001	nov
	talk-MULT → inspect → save → inspect → save	0.74	0.45	0.99	0.57	<.001	nov
	talk-MULT → inspect → save → inspect	0.74	0.45	0.99	0.57	<.001	nov
	start assessment → talk-MULT → inspect	0.51	0.26	0.51	0.26	<.001	nov
	talk-MULT → inspect → save	0.85	0.62	1.30	0.87	<.001	both
	inspect → save → inspect → save → inspect	0.82	0.60	1.83	1.18	<.001	both
	save → inspect → save → inspect → save	0.82	0.60	1.82	1.18	<.001	both
	inspect → save → inspect → save → inspect → save	0.82	0.59	1.81	1.17	<.001	both
	save → inspect → save → inspect	0.83	0.60	1.99	1.31	<.001	both

higher s-support and a significantly higher average i-support for novice students than experienced students.

Table 4 presents the top five differentially frequent long patterns that were common to both groups and the top five that were frequently used only by the novice group within each scenario. Most of these long sequential patterns entailed the repetition and combination of actions including inspecting objects, saving objects to backpack, discarding objects, and talking with NPCs. It seemed that novice students who had not used VPA before executed more sequences comprised of exploratory behaviors such as talking with NPCs and collecting data, while more experienced students focused primarily on what was necessary to answer the core inquiry question.

6. DISCUSSION AND CONCLUSION

This paper investigates the transfer of student science inquiry skills across two Virtual Performance Assessment scenarios, and the impact of the novelty of the immersive virtual environment on motivation and learning. We do so by comparing performance and behaviors between novice students and experienced students. A novelty effect was found as novice students who engaged in VPA for the first time showed significantly higher scores on motivational survey subscales such as interest/enjoyment, effort/importance, pressure/tension, value/usefulness, presence/immersion, and autonomy than more experienced students. As these students were first introduced to the novel 3D virtual environment, the initial attraction and attention led to higher enjoyment, greater effort invested in the tasks, a higher sense of immersion and a higher sense of autonomy. These measures tended to decline when students became relatively experienced and familiar with the environment, consistent with previous findings on the novelty effect [8, 12]. Sequential pattern mining and comparison of overall behavior prevalence using student action log data indicated that novice students engaged in more exploratory behaviors -- they collected more data in the environment and had higher frequency of long sequences comprised of exploratory actions such as talking with NPCs, manipulating objects, and collecting data, as compared to more experienced students. This, again, might be attributed to the novelty effect [cf. 14]. That is, the higher attention of novice students resulted in higher interest and efforts in exploring the new learning environment than students who were more experienced with VPA.

However, another possibility is that the experienced students focused more on the goal at hand, than on the environment they were researching this issue on, leading to less exploration and more attention directly to the information most likely to be useful. This itself may reflect the fact that novelty is wearing off, but may be a positive aspect of the disappearance of the novelty effect. Indeed, despite the experienced students' relatively lower motivation and fewer exploratory behaviors, they outperformed the novice students in identifying a correct final claim in both scenarios and in designing causal explanations (in one scenario). Experienced students generally showed more effective problem solving. They not only tended to read research information pages more often immediately after submitting a final claim or reviewing laboratory test results, but also spent more time reading the information each time they accessed a new page. As such, even after just a half hour completing the first assessment, students demonstrated more expert-like science inquiry behaviors -- they made more use of the research information available as resources [cf. 9], in order to either interpret results, or to monitor and reflect on their final claims [cf. 15]. The information from the pages may also have added to the domain-specific knowledge base of experienced students, which have been found to be crucial for problem solving and expertise development [5]. This corresponds to earlier findings that the transfer of domain-general inquiry strategy has the potential to facilitate the acquisition of domain-specific knowledge [4]. In conclusion, the experienced students successfully consolidated and transferred science inquiry skills they had learned from the first scenario during the approximately 30-minute engagement to the second scenario.

The current study contributes to research on the assessment of the transfer of science inquiry skills by proposing the application of a combination of educational data mining techniques such as sequential pattern mining as supplements to the traditional analysis of success between conditions. One limitation of this study is that the comparison conducted here involved virtual scenarios within the same VPA architecture. The fact that the two scenarios were highly structurally similar might have facilitated transfer. Future work may involve exploring whether far transfer of science inquiry occurs from VPA to assessments outside the system (e.g., other computer-based learning environments with different domain and interaction design).

7. ACKNOWLEDGMENTS

The research presented here was supported by the Bill and Melinda Gates Foundation. We also thank Chris Dede for his support and suggestions.

REFERENCES

[1] Agrawal, R., & Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the 11th IEEE International Conference on Data Engineering* (Mar. 1995), 3-14.

[2] Baker, R.S.J.d., Clarke-Midura, J. 2013. Predicting successful inquiry learning in a Virtual Performance Assessment for science. In *Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization*, 203-214.

[3] Bazaldua, D. A. L., Baker, R. S., San Pedro, M. O. Z. 2014. Combining expert and metric-based assessments of association rule interestingness. In *Proceedings of the 7th International Conference on Educational Data Mining*, 44-51.

[4] Chen, Z., & Klahr, D. 1999. All other things being equal: acquisition and transfer of the control of variables strategy. *Child Development,* 70(5), 1098-1120.

[5] Chi, M. T. H., Glaser, R., & Rees, E. 1982. Expertise in problem solving. In *Advances in the Psychology of Human Intelligence*, R. Sternberg, Ed. Vol. 1, Erlbaum, Hillsdale, NJ, 7–76.

[6] Clark, R. E. 1983. Reconsidering research on learning from media. *Review of educational research*, *53*(4), 445-459.

[7] Clarke-Midura, J., & Dede, C. 2010. Assessment, technology, and change. *Journal of Research, Education and Technology*, 42(3), 309-328.

[8] Cuban, L. 1986. *Teachers and Machines: The Classroom Use of Technology since 1920*. Teachers College Press, New York, NY.

[9] Gilhooly, K. J., McGeorge, P., Hunter, J., Rawles, J. M., Kirby, I. K., Green, C., & Wynn, V. 1997. Biomedical knowledge in diagnostic thinking: the case of

electrocardiogram (ECG) interpretation. *European Journal of Cognitive Psychology*, *9*(2), 199-223.

[10] Gutierrez-Santos, S., Mavrikis, M., & Magoulas, G. 2010. Sequence detection for adaptive feedback generation in an exploratory environment for mathematical generalisation. In *Artificial Intelligence: Methodology, Systems, and Applications*. Springer Berlin Heidelberg, 181-190.

[11] Hahsler, M., Gruen, B., & Hornik, K. 2005. Arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15).

[12] Keller, J. M. 1999. Using the ARCS motivational process in computer-based instruction and distance education. *New Directions for Teaching and Learning*, *1999*(78), 37-47.

[13] Kinnebrew, J. S., Loretz, K. M., & Biswas, G. 2013. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, *5*(1), 190-219.

[14] Kubota, C. A., & Olstad, R. G. 1991. Effects of novelty-reducing preparation on exploratory behavior and cognitive learning in a science museum setting. *Journal of research in Science Teaching*, *28*(3), 225-234.

[15] Kuhn, D., & Pease, M. 2008. What needs to develop in the development of inquiry skills? *Cognition and Instruction*, 26(4), 512-559.

[16] Kuhn, D., Schauble, L., & Garcia-Mila, M. 1992. Cross-domain development of scientific reasoning. *Cognition and Instruction*, 9, 285-327.

[17] Merceron, A., & Yacef, K. 2008. Interestingness measures for association rules in educational data. *Educational Data Mining*, 8, 57-66.

[18] National Research Council. 2011. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. The National Academies Press, Washington, DC.

[19] Ryan, R., Rigby, C., & Przybylski, A. 2006. The motivational pull of video games: A self-determination theory approach. *Motivation & Emotion, 30(4),* 344-360.

[20] Sabourin, J., Mott, B., & Lester, J. 2013. Discovering behavior patterns of self-regulated learners in an inquiry-based learning environment. In *Artificial Intelligence in Education* (Jan. 2013), Springer Berlin, Heidelberg, 209-218.

[21] Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., & Nakama, A. 2013. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction, 23*, 1-39.

[22] Sao Pedro, M., Jiang, Y., Paquette, L., Baker, R.S., Gobert, J. 2014. Identifying transfer of inquiry skills across physical science simulations using educational data mining. *Proceedings of the 11th International Conference of the Learning Sciences*, 222-229.

[23] Scheuer, O., & McLaren, B. M. 2012. Educational data mining. In *Encyclopedia of the Sciences of Learning*. Springer US, 1075-1079.

[24] Schofield, J. W. 1995. *Computers and Classroom Culture*. Cambridge University Press, New York, NY.

[25] Storey J. 2002. A direct approach to false discovery rates. *J Roy. Stat. Soc.*, 64, 479-498.

[26] Unity Technologies. 2010. *Unity Game Engine*.

[27] University of Rochester. 2015. *Intrinsic Motivation Inventory.* Retrieved January 25, 2015, from http://www.selfdeterminationtheory.org/intrinsic-motivation-inventory/

The Impact of Incorporating Student Confidence Items into an Intelligent Tutor: A Randomized Controlled Trial

Charles Lang
Harvard Graduate School of Education
13 Appian Way
Cambridge, MA
+1-617-495-7945
charles_lang@mail.harvard.edu

Neil Heffernan
Computer Science Department, Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
+1-508-831-5569
nth@wpi.edu

Korinn Ostrow
Learning Sciences & Technologies, Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
+1-508-831-5569
ksostrow@wpi.edu

Yutao Wang
Computer Science Department, Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
+1-508-831-5569
yutaowang@wpi.edu

ABSTRACT

For at least the last century researchers have advocated the use of student confidence as a form of educational assessment and the growth of online and mobile educational software has made the implementation of this measurement far easier. The following short paper discusses our first study of the dynamics of student confidence in an online math tutor. We used a randomized controlled trial to test whether asking students about their confidence while using an Intelligent Tutor altered their performance. We observe that (1) Asking students about their confidence has no statistically significant impact on any of several performance measures (2) Student confidence is more easily reduced by negative feedback (being incorrect) than increased by positive feedback (being correct) and (3) confidence accuracy may be a useful predictor of student behavior. This paper demonstrates how psychological ideas can be imported into Educational Data Mining and our findings point to the possibility of using student confidence to better predict performance and differentiate between students based on the way they approach items.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Psychology

K.3.1 [**Computers and Education]:** Computer-Assisted Instruction (CAI)

General Terms

Experimentation, Human Factors

Keywords

Confidence, certainty, self-efficacy, cognitive tutor, confidence-based assessment, ASSISTments

1. INTRODUCTION

Interest in student confidence arose out of investigations into the mathematical formalization of subjective probability at the end of the 19th century [5]. At least since 1913 researchers sought to apply these theories of judgment to educational assessments [19]. The initial motivation from the educationalists' perspective was to determine if querying student confidence could provide useful additional information about student performance [4]. Over the last century the utility of confidence testing has been demonstrated in terms of test reliability [3, 11, 15], identifying guessing [18], separating students based on their level of understanding [7], increasing student understanding [4, 6, 14] and explaining answer changing [17]. Interest in student confidence has been further extended through work on self-efficacy – "students' judgments of their capability to accomplish specific tasks" [1]. Self-efficacy studies have made extensive use of Likert-style questions about student confidence [12].

Despite the utility of student confidence it has not gained widespread use within educational assessment. This may be because experimental psychology largely views confidence as an unreliable measure, suggesting that humans generally tend to suffer from overconfidence bias [10]. Overconfidence bias implies that much of the variation in student confidence can be explained by an inclination for students to report that they are better at solving problems than they in fact are rather than explanatory variables that might improve learning [7].

Another reason for the failure of student confidence to become a widespread measure may be that the cost and logistical difficulty in collecting, scoring and storing confidence data was historically high. The comparatively low cost and large scale of online assessment may be diminishing this issue substantially though. In a world of yearly or bi-yearly paper tests it is not feasible to collect and score confidence data, but in an online environment these burdens are lifted.

Yet, there remain some lingering misgivings about the use of self-reported confidence. Overconfidence bias may be an artifact of larger issues with the way that confidence data are collected. Indeed, the concern remains whether simply asking students about their confidence may in fact alter their performance [13]. If requiring students to report their confidence reduces their overall performance then any utility in the measure will be undermined, it is therefore important to study the impact of student confidence measurement within a real-life setting.

The dynamics of student confidence are what concern this short paper. We were concerned primarily with the impact of asking Likert-style confidence questions on other aspects of student performance, and how students' confidence changed as they navigated tasks within the ASSISTments Intelligent Tutoring System. We are in the beginning stages of mapping out how student confidence changes as students move through online math assessment. Our aim is to identify how student confidence might relate to student behavior with the goal of leveraging this information to increase student learning.

2. METHOD

2.1 Data

The present study was conducted as a simple randomized controlled trial within ASSISTments, an adaptive mathematics tutor that serves as a free assistance and assessment tool to over 50,000 users around the world [9]. Two problem sets were designed around the multiplication and division of fractions and mixed numbers, using a mastery learning based structure called a Skill Builder. Skill Builder problem sets are unique in that students are randomly dealt questions from a skill bank until they are able to answer three consecutive questions accurately, thus 'mastering' the assignment.

Both problem sets were designed with two conditions: an experimental condition in which students were asked to self-assess their confidence is solving similar problems, and a control condition in which students were asked filler questions to control for the effect of spaced assessment. Random assignment was performed by the ASSISTments tutor at the student level. Throughout the course of each assignment, students were asked up to three self-assessment or survey questions. At the start of each assignment, students who were randomly assigned to the experimental condition were introduced to the skill of self-assessment, shown a set of problems isomorphic to those in the problem set, and asked to gauge their confidence in solving the problems using a Likert scale ranging from 'I cannot solve these problems (0%)' to 'I can definitely solve these problems (100%)'. Students who were randomly assigned to the control condition were polled on their current browser in an attempt to 'improve the ASSISTments tutor.' Examples of the initial questions posed to each condition are presented in Figure 1 below.

Following these initial questions, students were given three mathematics questions. If students solved each of these three questions accurately, the assignment was considered complete. However, if students answered at least one of the problems incorrectly, they would reach another self-assessment or survey question before being given another set of three math questions to try to master the problem set. This pattern happened a third time for students who were struggling with the content, until finally removing the self-assessment or survey element and simply providing back to back math questions until the student could solve three consecutive problems. Based on this design, high performing students were asked to gauge their confidence only a single time, while students struggling with the topic were asked to reassess their confidence up to two more times throughout the problem set. The confidence question was always formatted using the same Likert scale, while the 'ASSISTments' improvement surveys changed slightly, polling students on various elements of accessibility.

These Skill Builders were marked as ASSISTments Certified material and made publicly available to all users. The sets were promoted as new content and received high usage over the course of approximately three months. The tutor logged all student actions throughout the course of the experiment, and a dataset was obtained from the ASSISTments database for analysis. The experiment is still actively running within ASSISTments, gaining sample size for additional analysis to be conducted at a later time.

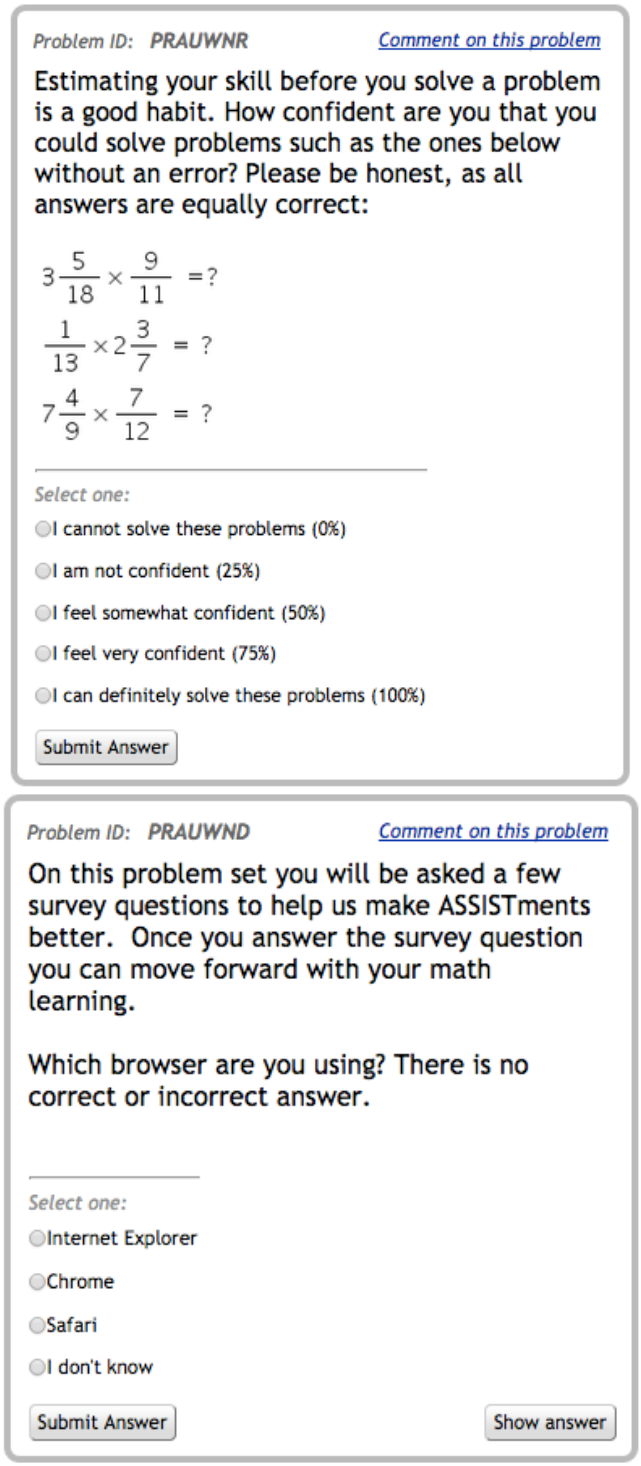


Figure 1. Initial Questions for Students in Experimental (Above) and Control (Below) Conditions

The data set used for the present analysis consisted of 950 12-14 year old students in the eighth grade, from a group of school districts in the North East the United States. Data included 10,770 problem level records including rich details pertaining to student performance. After working with the ASSISTments team to design and run this study, the lead author was provided the data set for primary analysis with all information that could lead to the identification of individual students removed, as set in the

protocol of an IRB exemption granted by the CUHS of Harvard University.

3. RESULTS

3.1 Student Confidence

3.1.1 Description of Confidence

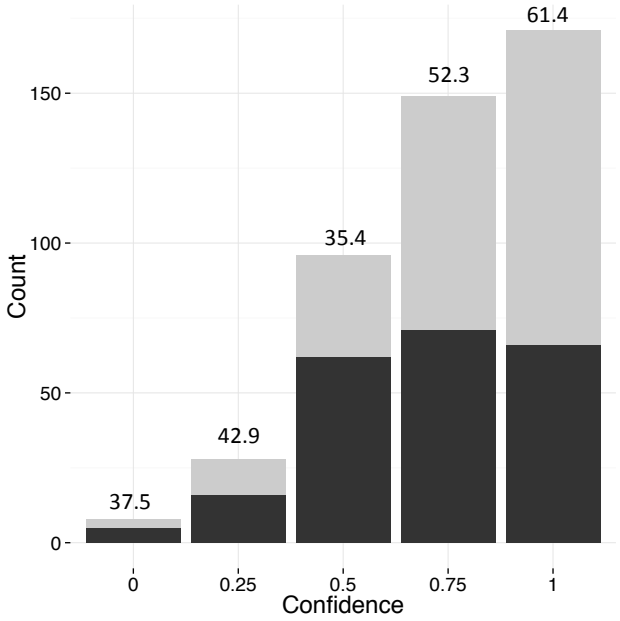


Figure 2. Histogram defining distribution of initial student confidence with the proportion of each group that was correct on the first item above the bar and shaded (gray:correct, black:incorrect). Most students have mid- to high-confidence.

The initial distribution of student confidence was left skewed, with the majority of students reporting their initial confidence in the problems as being between 0.5 and 1.0 (M = 0.75; Figure 2). On subsequent confidence questions the distribution remains left skewed though the mean confidence shifts toward the center as highly confident students exit the system after mastery (M = 0.56).

The overall trend in students' estimation of their own skill is that more of the confident students tend to be correct. However, the students at either extreme (not confident at all and 100% confident) do not meet their own expectations. Three of the eight students who estimated that they "cannot solve these problems" were able to solve the first problem and 66 out of the 105 students who estimated that they "can definitely solve these problems" were incorrect on the first problem.

3.1.2 Learning Gains

Overall learning gains were comparable between the experimental and control groups (Table 1). Though differences among different levels of confidence persisted. Highly confident students tended to be more accurate than the control group and continue to improve, while moderately to very unconfident students tended to be far less accurate than the control group, though they tended to improve, with the exception of the students with zero confidence. As occurred in the first question, those students who were "not confident" outperformed students who were "somewhat confident" on the second and third questions.

Table 1. Learning paths for students in the experimental and control groups showing percentage of students who were correct on questions 1, 2 and 3.

	Confidence					Treat	Control
	0.0	**0.25**	**0.5**	**0.75**	**1.0**		
Q1 Correct (%)	37.5	42.9	35.4	52.3	61.4	51.3	45.4
Q2 Correct (%)	62.5	60.7	55.2	68.5	76.6	68.1	70.7
Q3 Correct (%)	37.5	64.3	59.4	73.2	78.4	71.0	70.7
n	8	28	96	149	171	452	498

3.2 The Impact of Measuring Confidence on Performance

Since there is some evidence that question format can impact student performance we looked at whether there was a difference between students who were asked confidence style questions and those who were asked "dummy" survey questions. In all but one respect there seems to be no statistically significant effect of asking students what their confidence is within the ASSISTments system.

There was no statistically significant difference with respect to accuracy between students who were asked confidence questions and those who were not (Control = 53% correct, Experimental = 52% correct, χ^2 = 5.7, p = 0.68). Students who were asked confidence questions did not use more or less hints (Control = 0.89 hints/student, Experimental = 0.89 hints/student, χ^2 = 37.1, p = 0.09) nor did they make more or fewer attempts (Control = 1.7 attempts/student, Experimental = 1.6 attempts/student, χ^2 = 46.4, p = 0.41). There was also no difference between students who were asked about their confidence and those who were not with respect to the number of questions they answered (Control = 5.1 questions/student, Experimental = 5.2 questions/student, χ^2 = 169.7, p = 0.10). Nor did asking confidence questions impact the way that students behaved after being incorrect; there is no statistically significant tendency for students who were given confidence questions to ask for hints on the next question after being incorrect on the first question (Control = 8%, Experimental = 10%, χ^2 = 0.11, p = 0.74).

There is one case in which there is a statistically significant difference between the control and experimental groups though: of the students who were incorrect on the first question, more students in the experimental group were incorrect on the second question (χ^2 = 4.63, p = 0.03: Table 2). This suggests that the act of asking confidence questions impairs students' performance in some way. This effect disappears by the third question though (χ^2 = 0.61, p = 0.43).

Table 2. Students who were correct on Question 2 after being incorrect on Question 1 for control and experimental groups. Fewer students in the experimental group were correct on Question 2.

	Control	Experimental
Correct (%)	171 (34.3)	125* (27.7)
Incorrect (%)	327 (65.7)	327 (72.3)

* Denotes a significant difference between control and experimental $p < 0.05$.

3.3 The Importance of Confidence

3.3.1 Confidence as a Prediction of Future Performance

If we consider confidence to be a student's prediction of their future performance we can calculate an error measure of this prediction. For example, if a student has a confidence of 0.75 we would assume that they expected to get 75% of the next three questions correct. If they in fact got 100% of the answers correct then their error rate would 0.25 (confidence – percent correct).

Error rates appear to correlate with several factors, including accuracy. Students who are better at predicting their score on the next three questions tend to be those who are more accurate at answering those three questions ($r(452) = -0.54$, $p < .001$; Figure 3). They also tend to utilize more hints ($r(452) = 0.42$, $p < .001$) and make more attempts $r(452) = 0.31$, $p < .001$).

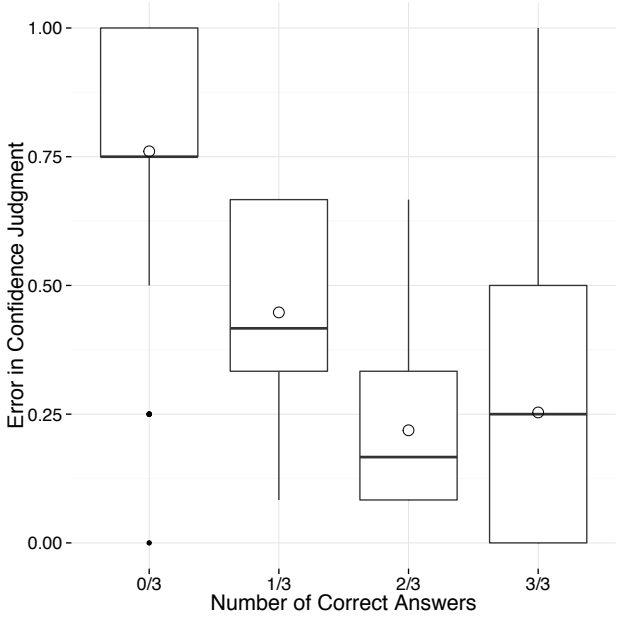


Figure 3. Boxplot representing the error associated with student confidence judgment (confidence – percent correct) vs. percent correct for first three questions. Students who are more accurate at judging their ability tend to get more answers correct. Line equals median, circle equals mean.

3.3.2 Predicting Accuracy Based on Confidence

We can also attempt to predict the outcome of a single question based on student confidence. We built a logistic regression model that predicted whether or not a student was correct on their third item using 1) student confidence, 2) whether the student was correct on previous items, 3) their percentage correct over all problem sets attempted, 4) how many problems they had attempted within the ASSISTments system, and 5) which problem set they were attempting. Of these predictors, the only significant variables were accuracy on previous questions and student confidence, which make up the most parsimonious model (Model IV; Table 3).

There is a more substantial relationship between accuracy on the third item and student confidence than with accuracy on the previous two items. A change in student confidence from zero to 100 is associated with the odds of being correct on the third question increasing by a factor of 3, whereas the odds of being correct on item 3 are increased by a factor of 2.3 with respect to being correct on the first item, and only 1.8 for being correct on the second item.

Table 3. Taxonomy of logistic regression models that display the fitted relationship between the log odds of being correct on the third item and student confidence, being correct on the first item, being correct on the second item, the prior percent correct, number of prior problems attempted and the problem set (n=452). Model IV is the most parsimonius.

	Model I	Model II	Model III	Model IV
Intercept	0.5688	-0.4254	-0.5359	-0.6684*
Confidence	0.9974*	1.2248**	1.3163**	1.1234**
Q1 Correct	0.7896***	0.9348***		0.8294***
Q2 Correct	0.6132**		0.7314***	0.5662***
Prior percent correct	-0.0001			
Prior problem count	0.3542			
Problem set	-0.0545			
AIC	517.75	518.3	526	514.15

3.3.3 Changes in Confidence after Incorrect Answers

The impact of incorrect answers on student confidence is clear from a breakdown of how confidence changes before and after completing questions (Figure 4). Students were asked for their confidence before the first and after the third problem. The decision tree below represents the 258 students who did not exit the system before they were asked this second round of

confidence questions. The tree is read top to bottom, in the first tier students are sorted based on how many of the three problems they got correct. In the second tier students are sorted based on how they changed their confidence, did they become less confident, more confident or stay the same.

There are a few trends that can be drawn out from this map. The majority of students (85%) who get three questions incorrect in a row lose confidence, while only 47% of students who get three correct in a row increase their confidence or are already at the maximum confidence. Indeed, 28% of students revise their confidence down after getting three correct answers in a row! Only one student decided to increase their confidence despite getting three incorrect answers in a row.

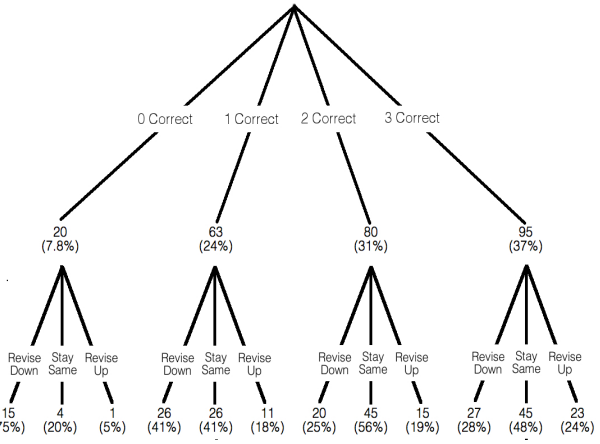


Figure 4. Changes in student confidence with respect to confidence levels at Question 1 and Question 5.

4. DISCUSSION

Overall the current study illustrates the trade off between using a different question format and the impact of this format on student behavior. Confidence style questions may provide substantial new utility in predicting and understanding student behavior but this utility may also come at a cost. We want to ensure that we have weighed this cost against the benefits of confidence style questions before further pursuing the benefits they provide. Overall, it appears from the present study that the benefits indeed do outweigh the costs.

4.1 Cost vs. Benefit

Beyond the time-cost of adding confidence questions to the problem set we wanted to know if there was any detrimental or beneficial impact on students performance of answering this kind of question and whether the question generates useful information.

The addition of Likert-style confidence questions appears not to impact many relevant behaviors within the ASSISTments system. This is somewhat surprising given methodological research on the impact of phrasing questions [16] and the substantial literatures on the impact of self-efficacy [12] and self-reflection [2] on student performance. However, in this study it seems to have had little discernable impact. The small impact that was detected however is of substantial concern. It appears that students who were given confidence style questions and who were incorrect on their first answer were slightly less likely to be correct on the second question they answered. We might imagine that asking students their confidence could have myriad effects on the way they answered, perhaps it made them more hesitant or more anxious resulting in poorer performance. In either case this is problematic as the aim of the system is to improve performance and learning.

This is not a definitive finding however, as the effect was small and disappeared by the next question. There are also alternative interpretations. The dip in performance may not necessarily connote a failure to learn. Perhaps it denotes a student wrestling more substantially with the concepts in the problem set, which may result in longer lasting, more robust learning going forward. This hypothesis needs to be tested by looking at future student performance. We also need to test whether any impact diminishes with exposure to the format.

Another reason why we may not want to use confidence style questions is that the information they generate is not useful because it is a poor estimate of student ability. We have substantial evidence of this conclusion. Students appear to be poor estimators of their own skill. For example, although unconfident students answer questions incorrectly more often than confident students, students at the extremes tend to exaggerate their predictions. Students with very low confidence tended to underestimate their ability and students with very high confidence tended to overestimate their ability. This trend may reflect how students approach confidence, although we have presented it as a continuous scale some students may be seeing it more as a binary; they are either confident of not. This would explain why very confident students get wrong answers and very unconfident students get correct answers and is in keeping with the psychological theory of extremeness [8]. In this theory people are thought to concentrate on the extremeness of options above all else. Therefore, students who maybe somewhat confident are drawn to concluding that they are either 0% or 100% confident. To conclude that there is no useful information in confidence because of this tendency would be a mistake though. There are two substantially useful characteristics that are worth pursuing within the ASSISTments system: error rate of student confidence and how confidence changes as students answer questions correctly or incorrectly.

Although students are, on average, poor judges of their own accuracy those who are better at predicting their accuracy tend to be more correct. There seems to be a benefit in being a good predictor of your own performance. This suggests the skill to predict your own performance may be a worthwhile cultivating and therefore measuring. This prediction skill is also correlated with higher levels of engagement with the system when a student is incorrect; asking for more hints and making more attempts. This may indicate that students who are better predictors of their own performance are also more interested in learning. This may help in signaling those students who are not interested in learning for differentiated interventions.

It is also worth thinking about how prediction accuracy is developed. The dynamics of confidence behavior can shed more light on this idea. Confidence seems to be very sensitive to accuracy in an interesting way. The vast majority of students who get incorrect answers tend to reduce their confidence, while a minority of students who get all answers correct seem to increase their confidence. Confidence, it would seem, is easier to lose than to gain. This may be related to another psychological principle, asymmetry. The asymmetry principle states that humans have a tendency to attribute greater weight to negative, rather than positive events. If this effect is cumulative it may explain why

students underestimate their ability at the low end of the confidence scale. Yet it doesn't explain why students overestimate their ability at the other end. Clearly there is more to understand about how students revise their confidence and the rate at which they do it. If being accurate in the prediction of your own performance is important, perhaps we should be more sensitive in how we impact that through the delivery of incorrect/correct answers. Perhaps pushing students away from extreme values is a worthwhile pursuit.

It would appear though that the benefits of studying confidence within this Intelligent Tutor far outweigh the possible cost of diminishing performance on one question. The ability to detect, and possibly increase, student engagement would be a highly useful addition.

4.2 Conclusion

The aim of this work is to develop understanding that can improve learning outcomes. It is useful information to know that student confidence is easier to reduce than to build and that accuracy in predicting ones performance is related to engagement in the system and increased performance. This can inform the way that difficulty is used to drive instruction, possibly balancing the difficulty and timing of problems with respect to student tolerances. In future research we hope to draw on the conclusions we have outlined here and to utilize associations with student confidence. In particular, we wish to investigate whether it is possible to improve students' estimates of their confidence and whether this translates into impact on their actions within the online tutor. We wish to know whether it is possible to increase persistence and increase the appropriate use of hints by targeting students' ability to estimate their confidence.

6. ACKNOWLEDGMENTS

Our thanks to the ASSISTments team for making this study possible. We acknowledge funding from multiple NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736 & DRL-1031398), the U.S. Department of Education (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation.

7. REFERENCES

[1] Bandura, A. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*. 84, 2 (1977), 191–215.

[2] Van den Boom, G., Paas, F., van Merriënboer, J.J.G. and van Gog, T. 2004. Reflection prompts and tutor feedback in a web-based learning environment: effects on students' self-regulated learning competence. *Computers in Human Behavior*. 20, 4 (Jul. 2004), 551–567.

[3] Ebel, R.L. 1965. *Measuring educational achievement*. Prentice-Hall.

[4] Echternacht, G. 1972. The use of confidence testing in objective tests. *Review of Educational Research*. 42, 2 (1972), 217–236.

[5] Estes, W.K. 1976. The cognitive side of probability learning. *Psychological Review*. 83, 1 (Jan. 1976), 37–64.

[6] Gardner-Medwin, A. and Gahan, M. 2003. Formative and summative confidence-based assessment. *Proceedings of the 2008 International Computer Assisted Assessment (CAA) Conference* (London, 2003), 147–155.

[7] Gardner-Medwin, A.R. 1995. Confidence assessment in the teaching of basic science. *ALT-J*. 3, 1 (Jan. 1995), 80–85.

[8] Griffin, D. and Tversky, A. 1992. The weighing of evidence and the determinants of confidence. *Cognitive Psychology*. 24, 3 (Jul. 1992), 411–435.

[9] Heffernan, N.T. and Heffernan, C.L. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*. 24, 4 (Sep. 2014), 470–497.

[10] Langer, E.J. 1975. The illusion of control. *Journal of Personality and Social Psychology*. 32, 2 (Aug. 1975), 311–328.

[11] Michael, J.J. 1968. The reliability of a multiple-choice examination under various test-taking instructions. *Journal of Educational Measurement*. 5, 4 (Dec. 1968), 307–314.

[12] Pajares, F. and David, M. 1994. Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*. 86, 2 (1994), 193–203.

[13] Pajares, F. and Urdan, T.C. 2006. *Self-efficacy Beliefs of Adolescents*. IAP.

[14] Ramsey, P.H., Ramsey, P.P. and Barnes, M.J. 1987. Effects of student confidence and item difficulty on test score gains due to answer changing. *Teaching of Psychology*. 14, 4 (1987), 206–210.

[15] Rippey, R. 1968. Probabilistic Testing. *Journal of Educational Measurement*. 5, 3 (Oct. 1968), 211–215.

[16] Schwarz, N. 1999. Self-reports: How the questions shape the answers. *American Psychologist*. 54, 2 (1999), 93–105.

[17] Skinner, N.F. 1983. Switching answers on multiple-choice questions: Shrewdness or shibboleth? *Teaching of Psychology*. 10, 4 (1983), 220–222.

[18] Taylor, C. and Gardner, P.L. 1999. An alternative method of answering and scoring multiple choice tests. *Research in Science Education*. 29, 3 (Sep. 1999), 353–363.

[19] Woodworth, R.S. 1915. *Archives of Psychology*.

Analyzing Early At-Risk Factors in Higher Education e-Learning Courses

Ryan S. Baker[1], David Lindrum[2], Mary Jane Lindrum[2], David Perkowski[2]
[1]Teachers College Columbia University, 525 W 120th St. New York, NY 10027
[2]Soomo Learning, 9 SW Pack Square, Suite 301, Asheville, NC 28801
baker2@exchange.tc.columbia.edu, david.lindrum@soomolearning.com, maryjane.lindrum@soomolearning.com, david.perkowski@soomolearning.com

ABSTRACT

College students enrolled in online courses lack many of the supports available to students in traditional face-to-face classes on a campus such as meeting the instructor, having a set class time, discussing topics in-person during class, meeting peers and having the option to speak with them outside of class, being able to visit faculty during office hours, and so on. Instructors also lack these interactions, which typically provide meaningful indications of how students are doing individually and as a cohort. Further, online instructors typically carry a heavier teaching load, making it even more important for them to find quick, reliable, and easily understandable indicators of student progress, so that they can prioritize their interventions based on which students are most in need. In this paper, we study very early predictors of student success and failure. Our data is based on student activity, and is drawn from courses offered online by a large private university. Our data source is the Soomo Learning Environment, which hosts the course content as well as extensive formative assessment. We find that students who access the resources early, continue accessing the resources throughout the early weeks of the course, and perform well on formative activities are more likely to succeed. Through use of these indicators in early weeks, it is possible to derive actionable, understandable, and reasonably reliable predictions of student success and failure.

Keywords

At-Risk Prediction, Prediction Modeling, Predictive Analytics, Activity Analytics, Online Course, Webtexts

1. INTRODUCTION

Students enrolled in online courses lack many of the supports available to students in traditional face-to-face classes on campus [13]. Drop rates are typically higher for online courses than traditional courses (see review in [8]), and procrastination is often a major problem in online courses [10]. Part of the reason for the lower success seen in online courses comes from the fact that faculty have less direct contact with students [5, 19] and as a result have fewer indicators of how students are doing, outside of formal assessment. This makes intervention for at-risk students more difficult than in campus-based learning settings.

As a result, many universities and providers of online courseware have moved to models that can automatically identify when students are at risk. These models identify indicators of potential student failure (or lower success). A comprehensive review of work in this area can be found in [10]. In one example of the creation and study of such a model, Barber and Sharkey [4] predicted course failure using a mixture of data from student finances, student performance in previous classes, student forum posting, and assignment performance. In a second example, Whitmer [17] predicted final course grade from student LMS usage activity, including the number of times a student accessed any content, the number of times a student read or posted to the forum, and the number of times a student accessed or submitted an assignment. In a third example, Romero and colleagues [15] predicted final course grade from activity and performance on assignments, including time taken by the student; this work was followed up by additional work, where the same group studied a more extensive set of interaction variables within the Moodle platform [14]. In a fourth example, Andergassen and colleagues [1] predicted final exam score from completion of online learning activities, including when in the semester students engaged those activities, and the total span of time between a student's first and last activities in the online resource.

An area of particular importance is early prediction, as recommended by Dekker and colleagues [7]. Being able to make predictions early in the semester, using the data available from initial student participation in the course, allows for timely intervention. There have been projects that have been successful in identifying at-risk students early in the semester. For example, Ming and Ming [12] developed models that could predict student course success from the first week of course participation, based on the topics students posted on the online discussion forum. In another example, Jiang and colleagues [11] predicted MOOC course completion from grades and discussion forum social network centrality, at the conclusion of the first course week.

Models that can predict student success early in a course, from course participation data, may be more or less useful depending on the features the models are based upon. If models are based on indicators which are interpretable and meaningful to course staff, these models can then provide instructors with data on which students are at-risk along with information on why those specific students are at risk. Systems of this nature have been successfully embedded within intervention practices and had positive impacts on student outcomes. For example, the Course Signals project at Purdue University provides predictions to instructors along with suggested interventions for specific students, in the form of recommended emails to send the students [2]. In one evaluation, Course Signals was associated with better student grades and better retention [3]. Another project, the Open Academic Support Environment, was associated with better student grades [10].

The attributes of a desirable predictive model are tightly connected to the potential uses of that model. For example, highly complex "black box" indicators are hard for instructors to use in interventions, even if they might be perfectly suitable for automated interventions. Beyond this, demographic variables (such as race and financial need) can be predictive [17, 18], but are less immediately useful for instructors wishing to intervene.

In this paper, we study early predictors of student success based on student activity, with the goal of giving faculty immediately

useful, easy-to-interpret data.

We analyze these predictors within the context of the Soomo Learning Environment, a system used by over 100 universities to deliver course content and extensive formative assessment to over 70,000 undergraduates a year. Specifically, in this paper we study the learning and eventual success of over four thousand students taking an online course on introductory history at a large 4-year private university.

We find that students who access the resources early, continue accessing the resources throughout the early weeks of the course, and perform well on formative activities are more likely to succeed in the course overall. Through use of these indicators in early weeks, it is possible to derive actionable, understandable, and reasonably reliable, predictions of student success, enabling faculty to identify those students most in need of intervention, and suggesting the kind of guidance each student needs.

2. DATA

We investigate these issues within the context of data from an introductory history course, offered as an online course by a large 4-year private university, using an interactive web-based learning resource from Soomo. The Soomo Learning Environment (SLE) is a web-based content management system built for hosting instructional content and formative assessment. Typically students click a link in their learning management system to open their webtext, hosted in the SLE, in a new tab. All course content, customized for the specific instructor and institution, is presented within this environment. Courses are typically built with a mix of original, permissioned, and open content, combining text, images, audio, video, hosted and linked artifacts, and tools for study. Webtexts are developed by instructional designers at Soomo Learning in conversation with faculty advisors and subject matter experts. Webtexts are then peer reviewed and finally tailored to the needs of a specific institution and/or faculty member.

Webtexts are not just digital copies of traditional paper textbooks; they are distinguished by hundreds of opportunities for students to respond to the content through the course Within Soomo's webtexts, "Study Questions" help students assess their own comprehension of what they just read or watched. "Investigations" present opportunities for application, analysis, synthesis, and evaluation, thereby supporting learners in developing richer understanding.

Final student grades in the US History course were based on performance on a range of assignments. The grade weighting was identical across sections in a specific term, but varied term-to-term as the university and Soomo Learning worked together to tune the course. The final course grade was based on a combination of a final paper and milestones to that final paper, work in the Soomo Learning Environment, and participation in class discussion boards. We obtained data on student course performance and webtext activity, for 4,002 students enrolled across 140 sections of this course, taught over six terms in 2013 and 2014. These students performed a total of 2,053,452 actions in the webtext, including opening pages and answering questions.

Student grades below 60% were considered failing grades; however, the target of our at-risk predictions was to predict whether students would fall below 73%, the minimum grade required to get a C. 990 of the 4,002 students (24.7%) obtained a grade below 73%.

3. ANALYZING INDIVIDUAL PREDICTORS

One of the major goals of predictive analytics is making predictions early in the semester, before the student has fallen behind on the course's material to an extent that is difficult to repair. It is at this stage where instructor intervention can have the greatest impact. In this paper, therefore, we focus on student performance and usage in the first 4 weeks of a 10-week term.

The Soomo webtexts include formative assessment throughout the course, starting on the first pages of the resource. This gives faculty measures of student engagement and performance from the very first week of the course. The predictors analyzed in this paper are not inherent to the Soomo Learning Environment – they could be applied to other online courses that have online readings and assignments. They rely primarily on having measures of student engagement and understanding on a regular basis, from the start of the course.

3.1 Did the student access the webtext at all?

The first feature we analyze is whether students accessed the webtext at all in the early stages of the course. This course was organized into a set of one-week units. Therefore, it might be plausible to analyze whether a student accessed the webtext during the first week of the course; by the end of the first week, the students were expected to have completed the first week's materials. However, many students procrastinate [16], and students are not penalized within this course for completing materials late, so it is possible that many students do not access course materials within this window. We analyze variants of this feature, looking at whether students have failed to access the webtext and activities within the first N days of the course. The canonical value of N is 7; other values are also examined. (We omit data from one course term for this analysis in specific, due to a logging error).

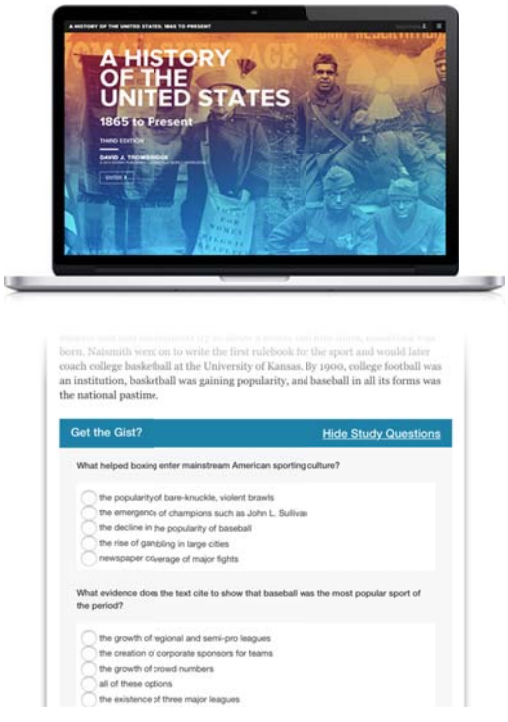


Figure 1. The introductory US History webtext (above) and embedded study questions relevant to that text (below)

As such, we predict whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student had accessed the book yet by day N. A precision-recall curve for this relationship is shown in Figure 2. A precision-recall curve [6] shows the tradeoff between precision and recall for different thresholds of a model. Precision represents the proportion of cases identified as at-risk that are genuinely at-risk; recall represents the proportion of genuinely at-risk cases that are identified as at-risk. They are computed:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Typically, precision-recall curves are used for different confidence thresholds between a positive and negative prediction; in this case, we display the tradeoff between precision and recall for different thresholds of how many days into a course a student can be before we become concerned that they have not accessed the webtext yet. As will be seen in the paper, studying these curves allows us to study the relative trade-off between precision and recall for different model thresholds and different feature variants. Some instructors may want models with higher recall, so that they can contact a larger proportion of at-risk students; other instructors may want more models with higher precision, to avoid contacting too many total students. While some researchers argue for optimizing a single metric, different instructors (or university administrators) may prefer different models.

As Figure 2 shows, there is a clear trade-off between precision and recall for how many days have passed at the start of the course without the student accessing the webtext. On the far left, almost all students who have not yet accessed the webtext by the 14th day of the class fail. On the far right, almost all students who eventually fail are captured by a model that looks at whether the student has not yet accessed the webtext seven days before class, but precision is only 40%. On the first day of class (day 0), precision is barely higher but recall is much lower. Seven days later (day 7), precision approaches 80% but recall is just below 20%. As such, this indicator changes its meaning considerably with each day that passes during the first 7 days of the class. On day 0, the Cohen's Kappa for this feature (representing the degree to which the model is better than chance) is 0.207. On day 7, Kappa is 0.200. On day 3, it reaches a maximum of 0.277; any value of N higher or lower than 3 has a lower Kappa.

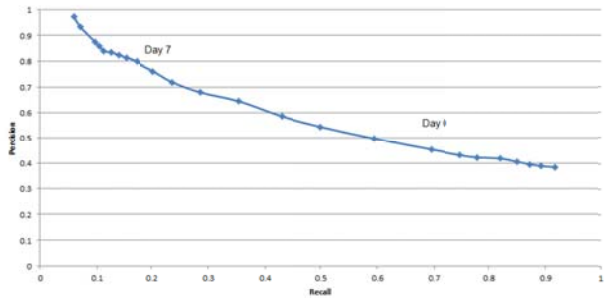


Figure 2. Precision-Recall Curve for how well a final grade below 73% is predicted by whether a student has accessed the webtext by day N.

3.2 Has the student accessed the webtext recently?

Accessing the webtext is an important first step, but it is reasonable to believe that students are most successful if they continue to access the course materials weekly. As such, the second feature we analyze is how long it has been since the student accessed the webtext. This feature has two parameters: the current day N, and the number of days D since the student last accessed the webtext.

As such, we are predicting whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student had accessed the book in the last D days, at the time of day N. For tractability, we select four possible values for D: the last 3 days, the last 5 days, the last 7 days, and the last 10 days. We also select values between 1 and 28 for N; the model does not go beyond the fourth week of this course, because after this point, it is relatively late for "early" intervention. Note that students can open the book before the first day of the course (so it is meaningful to compare between values of D, even for N=1).

A set of precision-recall curves is given for these model variants in Figure 3. As Figure 3 shows, the models start out very similar, regardless of value of D, at the beginning of the course, with precisions around 44%-46% and recalls around 65%-70%.

As the value of N goes up, recall drops and precision goes up, until the changes become unstable around the third week of the course. (At that point, however, the changes are relatively minimal). The higher the value of D, the higher the eventual precision and the lower the eventual recall, at the end of the fourth week of the course. For instance, for D = 7, the precision reaches 80.4% by day 14, though the recall is at a relatively low 16.7%. To put this another way, on day 14, a student who has not accessed the textbook in the last 7 days has a 80.4% probability of performing poorly in the course, and 16.7% of students who perform poorly in the course had not accessed the textbook in the last seven days on day 14.

This shift effect is relatively weaker for lower values of D; for instance, for D = 3, the precision goes up relatively little, reaching only 54.2% on day 4, while the recall drops rapidly, reaching 35.8% by day 7. These results, in aggregate, show that this feature manifests different behavior depending on choice of threshold.

Kappa values were relatively unstable across predictors, though the differences in Kappa were generally small, indicating that most of the differences between models reflected a precision-recall tradeoff. The best Kappa, 0.27, was obtained for D=7 and N=28. The second best kappa, 0.247, was obtained for D=7 and N=4. However, the third best kappa, 0.241, was obtained for D=3 and N=4. Kappa values were generally higher for higher values of D, but the differences were extremely small; the average Kappa for each value of D only varied by 0.03.

3.3 Is the student doing poorly on exercises in the webtext?

Another indicator that the student is struggling is if the student is performing poorly on the formative exercises in the webtext. These exercises comprise only a third of the student's eventual grade, but are an indicator that the student does not understand the content. As discussed above, there are two types of assignments within the webtext, Study Questions and Investigations.

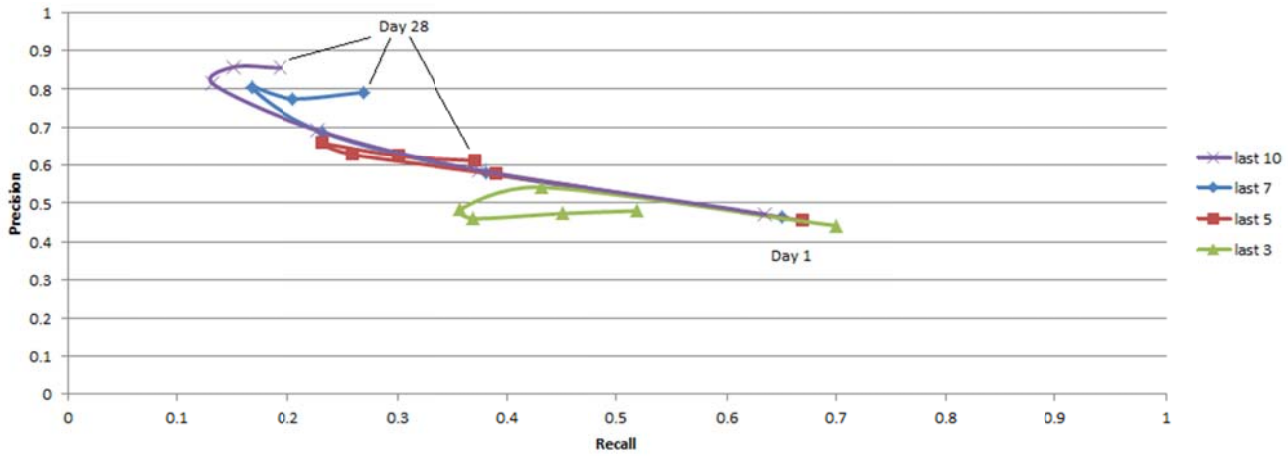


Figure 3: Precision-Recall Curve for how well a final grade below 73% is predicted by whether a student has accessed the webtext in the last D days (indicated by color), by day N.

We can look at student performance on these two types of assignments, first filtering out students who have not completed any assignments, and then looking for students who by the end of the first or second week of content (day N = 7 or 14) have an average below a cut-off S for Study Questions, and a cut-off I for Investigations. As such, we are predicting whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student averaged below S on Study Questions and I on investigate assignments, at the time of day N.

Optimizing based on Cohen's Kappa, and setting N = day 7, we find that the value of S has almost no impact (and are therefore not shown on Figure 3). For example, if the I cutoff = 70%, any value of S from 50% to 95% results in a Cohen's Kappa between 0.18 and 0.20. If the I cutoff = 85%, any value of S from 50% to 95% results in a Cohen's Kappa between 0.08 and 0.10.

By contrast, the value of I has substantial impact on model goodness. If the I cutoff = 65% (and S = I), Kappa is 0.20. If the I cutoff = 95% (and S=I), Kappa is -0.05.

The reason for this difference in predictive power between Study Questions and Investigations is likely that Study Questions can be reset. That is, when a student answers a set of Study Questions, the attempt is immediately graded. Students are given feedback and an opportunity to reset the questions and answer them again. Students are encouraged to do this in order to understand the correct answer before they move on. Investigations are more complex, and are also not resettable. In general, then, scores on Study Questions indicate effort and scores on Investigations indicate understanding.

Setting S = I, we can compute the precision-recall curve for different values of I, shown in Figure 4.

As Figure 4 shows, as the required grade to not be considered at-risk goes up, the recall goes up but the precision goes down, leading to very different models for different thresholds. It does not appear to make a big difference, however, whether we look at the first week of content, or the first two weeks of content.

To break this down, students who got below 95% on the first week of Soomo Learning Environment content had a 34.0% probability of performing poorly, and 81.8% of students who performed poorly in the course obtained below 95% on the first week of Soomo Learning Environment content. Students who got below 50% on

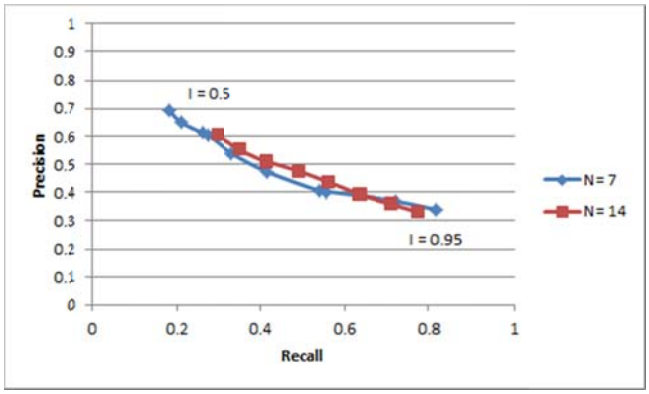


Figure 4. Precision-Recall Curve for how well a final grade below 73% is predicted by average grade on assignments (I), by day (N) 7 and 14.

the first week of Soomo Learning Environment content had a 69.5% probability of performing poorly, and 18.1% of students who performed poorly in the course obtained below 50% on the first week of Soomo Learning Environment content. As Figure 4 shows, the trade-off between precision and recall is roughly even for values of S and I between 50% and 95%.

4. INTEGRATED PREDICTIVE MODEL

Having computed these three indicators, it becomes feasible to look at the three in concert, to see how well we can do overall at predicting whether a student is at risk of obtaining a low grade.

The most straightforward way to do so would simply be to combine the single best version of the three operators described above, with an "or" function. Taking the students who obtained below 95% on the first week of Soomo Learning Environment content, the students who had not yet opened the book on day 2, and the students who had not accessed the book in the last 7 days on day 28, and combining them using an "or" function ends up with the prediction that 98.6% of students are at-risk, a model that is not very usable for intervention (the instructor intervenes for all students).

Alternatively, we can use higher-precision, lower-recall versions of these metrics. Taking the students who obtained below 50% on the first week of Soomo Learning Environment content, the students who had not yet opened the book on day 7, and the students who had not accessed the book in the last 3 days on day 7, and

combining them using an "or" function ends up with the prediction that 84.7% of students are at-risk, still too many interventions.

If, by contrast, we use "and" across the three operators, trying to find students who are definitely not at-risk (e.g. students who demonstrate none of the three behaviors that are indicative of an at-risk student), the higher-precision, lower-recall version of the metrics identifies exactly four students out of 4002 as being at risk. The lower-precision, higher-recall version of the metrics identifies 14.1% of the students as being at-risk, a more workable number for intervention. However, the model achieves a precision of 25.8% and a recall of 10.2%, much worse numbers than single-feature models.

An alternate approach, which we use in this section, is to use a machine-learned model to combine the features in a more complex way. In these analyses, we conduct cross-validation as a check on over-fitting, to determine how reliable these models will be for new students in future sections of the course. Given the focus on predicting performance for future course sections, we conduct the cross-validation at the grain-size of course sections.

We input to the models the best variants of each feature (in terms of Kappa) seen in the previous sections. We also input extreme threshold variants of the features (high precision-low recall and low precision-high recall) when they achieve comparable Kappa to the best variants. In specific, we include whether the student opened the book on the first N days after the course start (0 days, 2 days, 7 days), whether the student accessed the book recently (D=7, N=28; D=7, N=4; D=3, N=4), and performance on assignments (wk. 1 only, S=I=0.65).

We applied several classification algorithms to these features, and evaluated the resultant models using Kappa, precision, recall, and A', shown in Table 1. A' is the probability that the model can distinguish whether a student is in the at-risk category or not. A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly [9]. A' is used rather than the theoretically equivalent AUC ROC implementation, due to bugs in existing implementations of AUC ROC.

As is often the case, there is not a single best model across all metrics. The best A' is obtained by W-KStar; but this algorithm's Kappa is much lower than other algorithms with very similar A'. Arguably, Logistic Regression, with A' only 0.015 lower than W-Kstar, but Kappa 0.111 better, should be preferred. Logistic Regression also achieves the best Recall among the algorithms, while obtaining a middling Precision. Of course, it should be remembered that Recall and Precision can always be traded-off by selecting an alternate threshold based on a Receiver-Operating Characteristic curve, or a Precision-Recall curve (as used throughout this paper), shown in Figures 5 and 6. These curves

Table 1. Performance of Integrated Predictive Models.

Algorithm	Kappa	Precision	Recall	A'
W-J48	0.315	0.636	0.435	0.655
W-JRip	0.265	0.570	0.468	0.578
Naïve Bayes	0.231	0.532	0.483	0.666
W-KStar	0.233	0.670	0.288	0.677
Step Regression	0.305	0.697	0.353	0.658
Logistic Regression	0.344	0.568	0.595	0.662

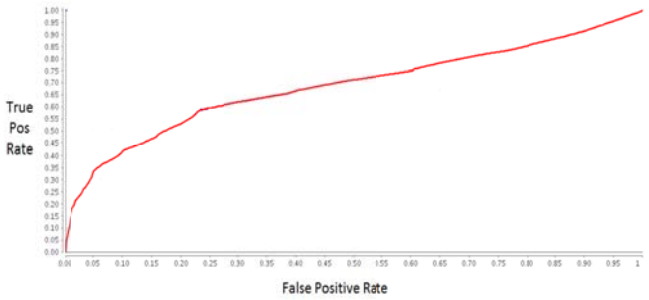


Figure 5. Receiver-Operating Characteristic Curve for (Cross-Validated) Logistic Regression Version of Integrated Predictive Model.

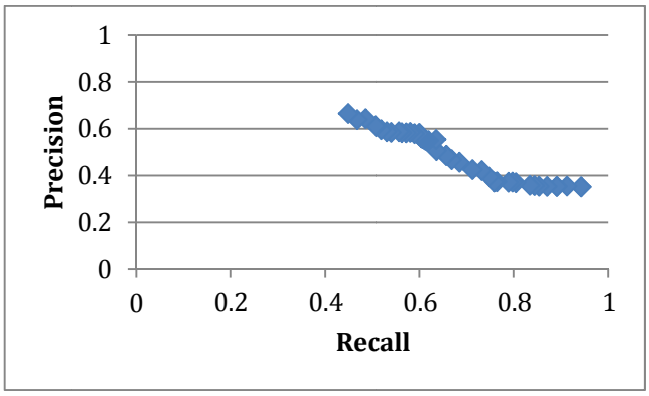


Figure 6. Precision-Recall Curve for (Cross-Validated) Logistic Regression Version of Integrated Predictive Model.

indicate that recall can be increased to 94.3%, while maintaining precision of 35.1%.

5. DISCUSSION AND CONCLUSIONS

In this paper, we have investigated the degree to which student participation in webtext activities within the Soomo Learning Environment, early in the semester, are predictive of eventual student success in a course. We find that it is indeed possible to achieve a reasonable degree of predictive power, and to identify a substantial proportion of the at-risk students, with reasonable precision. Some of these measures have predictive value from the first day of the course, allowing very early intervention.

In aggregate, we find that a combination of these measures leads to A' values in the 0.65-0.7 range, sufficient for intervention, though not quite up to the level of medical diagnostics. The logistic regression version of the combined model can identify 59.5% of students who will perform poorly, achieving precision of 56.8%, 34.4% better than chance. Of course, with any of the approaches used here, confidence thresholds for intervention can be adjusted, leading to more or fewer interventions. If high recall is the goal – attempting to provide intervention to most at-risk students even if some interventions are mis-applied – then the threshold of the logistic regression model can be adjusted, resulting in a model that can identify 94.3% of the students who will perform poorly, but where only 35.1% of the students it identifies performs poorly. This model does better than a single-feature model; even the high recall model from section 3-3 (performance under 95% on early assignments within the webtext) obtained a recall of 81.8% -- lower than the logistic regression model – while achieving comparable precision (34.0%).

However, if the goal is to provide high-cost interventions to the students who are very likely to perform poorly, the logistic regression model is not an optimal choice. The logistic regression model cannot achieve very high precision, even through adjusting thresholds, as shown in Figure 6. However, an alternate approach can be adopted, through using a different predictor algorithm, step regression. This algorithm obtains more precise prediction than logistic regression, with precision of 69.7% and recall of 35.3% for standard thresholds.

Importantly, these measures are based upon interpretable features. They are based upon features that instructors identified as meaningful and having the potential for intervention. The combination of individual-feature models and a comprehensive model enables us to identify which students are at risk, and then to provide instructors with information about which students are at risk, and why. We can specifically identify that a student is at risk because he/she has failed to access the resources, or because he/she has failed to complete the assignments on time, or because he/she has scored poorly on the assignments. With this information, automatically distilled and placed in a user interface within the Soomo platform, faculty will have a means of finding students who most need support and a basis for encouraging them to access the text, do the assigned work, and take the time to do it well.

The first area of future work planned is to enhance the analytics already offered to instructors by Soomo, based on the findings presented here. The success of these interventions, both in terms of improved student grades and improved student retention, will be evaluated in an experiment or quasi-experiment (the final study design will depend upon negotiation with the university which partnered on the analyses discussed in this paper).

However, beyond testing interventions based on the model presented here, there is considerable future work to extend, improve, and study the generalizability of these models. For example, it will be valuable to study what characterizes the students for whom this model functions less effectively. Can additional features, like how much time students spend on assignments, improve overall prediction? And how well will the features identified here apply for different courses, and for different universities, an issue explored by Jayaprakash et al. [10], among others. By answering these questions, we can improve the models, verify their broad applicability, and move to using the models within intervention strategies that can achieve broad positive impact on learners.

6. ACKNOWLEDGMENTS

This research was made possible by the active cooperation of our partner university.

7. REFERENCES

[1] Andergassen, M., Modrtischer, F., Neumann, G. (2014) Practice and Repetition during Exam Preparation in Blended Learning Courses: Correlations with Learning Results. *Journal of Learning Analytics, 1* (1), 48-74.

[2] Arnold, K. (2010) Signals: Applying Academic Analytics. *Educause Quarterly*. March 2010.

[3] Arnold, K., Pistilli, M. (2012) Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *Proc. of the 2nd International Conference on Learning Analytics.*

[4] Barber, R., Sharkey, M. (2012) Course Correction: Using Analytics to Predict Course Success. *Proceedings of the 2nd International Conference on Learning Analytics*, 259-262.

[5] Beard, L.A., Harper, C. (2002) Student Perceptions of Online versus on Campus Instruction. *Education, 122* (4).

[6] Davis, J., Goadrich, M. (2006) The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine learning.*

[7] Dekker, G., Pechenizkiy, M., Vleeshouwers, J.M. (2009) Predicting Students Drop Out: A Case Study. *Proc. of the 2nd Int'l. Conference on Educational Data Mining*, 41-50.

[8] Diaz, D.P. (2002) Online Drop Rates Revisited. *The Technology Source*, May/June 2002.

[9] Hanley, J.A. and McNeil, B.J. (1982) The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology, 143*, 29-36.

[10] Jayaprakash, S.M., Moody, E.W., Lauria, E.J.M., Regan, J.R., Baron, J.D. Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics, 1* (1), 6-47.

[11] Jiang, S., Williams, A.E., Schenke, K., Warschauer, M., O'Dowd, D. (2014) Predicting MOOC Performance with Week 1 Beahvior. *Proceedings of the 7th International Conference on Educational Data Mining*, 273-275.

[12] Ming, N.C., Ming, V.L. (2012) Automated Predictive Assessment from Unstructured Student Writing. *Proceedings of the 1st international Conference on Data Analytics.*

[13] Muilenburg, L.Y., Berge, .L. (2005) Student Barriers to Online Learning: a factor analytic study. *Distance Education, 26* (1), 29-48.

[14] Romero, C., Olmo, J.L., Ventura, S. (2013) A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets. *Proc. of the 6th Int'l. Conference on Educational Data Mining*, 268-271.

[15] Romero, C., Ventura, S., Garcia, E. (2007) Data mining in course management systems: Moodle case study and tutorial. *Computers and Education, 51* (1), 368-384.

[16] Thille, C., Schneider, E., Kizilcec, R.F., Piech, C., Halawa, S.A., Greene, D.K. (2014) The Future of Data-Enriched Assessment. *Research and Practice in Assessment, 9* (4), 5-16.

[17] Whitmer, J. (2012) *Logging on to improve achievement: Evaluating the relationship between use of the learning management system, student characteristics, and academic achievement in a hybrid large enrollment undergraduate course*. Unpublished Doctoral Dissertation, UC Davis.

[18] Wolff, A., Zdrahal, Z., Nikolov, A., & Pantucek, M. (2013). Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment. *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 145–149.

[19] Zhang, J., & Walls, R. (2006). Instructors' self-perceived pedagogical principle implementation in the online environment. *The Quarterly Review of Distance Education, 7*(4), 413–426.

Do Country Stereotypes Exist in PISA? A Clustering Approach for Large, Sparse, and Weighted Data.

Mirka Saarela
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
mirka.saarela@gmail.com

Tommi Kärkkäinen
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
tommi.karkkainen@jyu.fi

ABSTRACT

Certain stereotypes can be associated with people from different countries. For example, the Italians are expected to be emotional, the Germans functional, and the Chinese hard-working. In this study, we cluster all 15-year-old students representing the 68 different nations and territories that participated in the latest Programme for International Student Assessment (PISA 2012). The hypothesis is that the students will start to form their own country groups when clustered according to the scale indices that summarize many of the students' characteristics. In order to meet PISA data analysis requirements, we use a novel combination of our previously published algorithmic components to realize a weighted sparse data clustering approach. This enables us to work with around half a million observations with large number of missing values, which represent the population of more than 24 million students globally. Three internal cluster indices suitable for sparse data are used to determine the number of clusters and the whole procedure is repeated recursively to end up with a set of clusters on three different refinement levels. The results show that our final clusters can indeed be explained by the actual student performance but only to a marginal degree by the country.

Keywords

Weighted Clustering, PISA, Sparse Cluster Indices, Country Stereotype

1. INTRODUCTION

Certain stereotypes seem to be associated with people from different countries. The French and Italians, for example, are expected to be emotional, while Germany has mainly a functional country stereotype [4], and the Chinese are commonly perceived as hard-working [3]. According to the *Hofstede Model* [6], national cultures can be characterized along six dimensions: power distance, individualism, masculinity, uncertainty avoidance, pragmatism, and indulgence. The hypothesis in this study is that also the population of 15-year-old students worldwide will start to form their own national groups, i.e., show similar characteristics to their country peers, when clustered according to their attributes and attitudes towards education.

PISA (Programme for International Student Assessment) is a worldwide triannual survey conducted by the Organisation for Economic Co-operation and Development (OECD), assessing the proficiency of 15-year-old students from different countries and economies in three domains: reading, mathematics, and science. Besides evaluating student performances, PISA is also one of the largest public databases[1] of students' demographic and contextual data, such as their attitudes and behaviours towards various aspects of education.

In order to test our hypothesis, we utilize the 15 PISA scale indices (explicitly detailed in [14]), a set of derived variables that readily summarize the background of the students including their characteristics and attitudes. In particular, the *escs* index measures the students' economic, social and cultural status and is known to account for most variance in performance [9]. Additionally, 5 scale indices (*belong*, *atschl*, *attlnact*, *persev*, *openps*) are generally associated with performance on a student-level, while 9 further ones (*failmat*, *intmat*, *instmot*, *matheff*, *anxmat*, *scmat*, *mathbeh*, *matintfc*, *subnorm*) are directly related to attitudes towards mathematics, the main assessment area in the most recent survey (PISA 2012). However, since the assessment material exceeds the time that is allocated for the test, each student is administered solely a fraction of the whole set of cognitive items and only one of the three background questionnaires. Because of this rotated design, 33.24% of the PISA scale indices values are missing.

Moreover, PISA data are an important example of large data sets that include weights. Only some students from each country are sampled for the study, but multiplied with their respective weights they should represent the whole 15-year-old student population. The sample data of the latest PISA assessment, i.e., the data we are working with, consists of 485490 students which, taking the weights into account, represent more than 24 million 15-year-old students in the 68 different territories that participated in PISA 2012.

[1]See `http://www.oecd.org/pisa/pisaproducts/`.

The content of this paper is as follows. First, we describe the clustering algorithm that allows us to work with the large, sparse and weighted data (Sec. 2). Second, we present the clustering results (Sec. 3) and their relevance to our hypothesis, i.e., how the clusters on the different levels can be characterized and to what extent they form their own country groups. Finally, in Sec. 4, we conclude our study and discuss directions for further research.

2. THE CLUSTERING APPROACH

Sparsity of PISA data must be taken into account when selecting or developing a data mining technique. With missing values one faces difficulties in justifying assumptions on data or error normality [14, 15], which underlie the classical second-order statistics. Hence, the data mining techniques here are based on the so-called nonparametric, robust statistics [5]. A robust, weighted clustering approach suitable for data sets with a large portion of missing values, non-normal error distribution, and given alignment between a sample and the population through weights, was introduced and tested in [16]. Here, we apply a similar method with slight modifications, along the lines of [7] for sampled initialization and [17] for hierarchical application. All computations were implemented and realized in Matlab R2014a.

2.1 Basic method

Denote by N the number of observations and by n the dimension of an observation of the data matrix $\mathbf{X}$; and let $\{w_i\}, i = 1, \ldots, N$ be the positive sample-population-alignment weights. Further, let $\{\mathbf{p}_i\}, i = 1, \ldots, N$, be the projection vectors that define the pattern of the available values [10, 1, 14, 15]. The weighted spatial median $\mathbf{s}$ with the so-called available data strategy can be obtained as the solution of the projected Weber problem

$$\min_{\mathbf{v}\in\mathbf{R}^n} \mathcal{J}(\mathbf{v}), \quad \mathcal{J}(\mathbf{v}) = \sum_{i=1}^{N} w_i \|\mathrm{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{v})\|, \qquad (1)$$

where $\mathrm{Diag}\{\mathbf{p}_i\}$ denotes the diagonal matrix corresponding to the given vector $\mathbf{p}_i$. As described in [8], this optimization problem is nonsmooth, i.e., it is not classically differentiable. However, an accurate approximation for the solution of the nonsmooth problem can be obtained by solving the regularized equation (see [1]) $\sum_{i=1}^{N} \frac{w_i \mathrm{Diag}\{\mathbf{p}_i\}(\mathbf{s}-\mathbf{x}_i)}{\max\{\|\mathrm{Diag}\{\mathbf{p}_i\}(\mathbf{s}-\mathbf{x}_i)\|, \delta\}} = \mathbf{0}$ for $\delta > 0$. This is solved using the SOR (Sequential Overrelaxation) algorithm [1] with the overrelaxation parameter $\omega = 1.5$. We choose $\delta = \sqrt{\varepsilon}$ for ε representing the machine precision.

In case of clustering with K prototypes, i.e., the centroids that represent the K clusters, one determines these by solving the nonsmooth problem $\min_{\{\mathbf{c}_k\}_{k=1}^K} \mathcal{J}(\{\mathbf{c}_k\})$, where all $\mathbf{c}_k \in \mathbf{R}^n$ and

$$\mathcal{J}(\{\mathbf{c}_k\}) = \sum_{k=1}^{K} \sum_{i\in I_k} w_i \|\mathrm{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{c}_k)\|. \qquad (2)$$

Hereby, I_k determines the subset of data being closest to the kth prototype $\mathbf{c}_k$. The main body of the so-called iterative relocation algorithm for minimizing (2), which is referred as *weighted k-spatialmedians*, consists of successive application of the two main steps: i) find the closest prototype for each observation, and ii) recompute all prototypes $\mathbf{c}_k$ using the attached subset of data. For the latter part, we compute the weighted spatial median as described above. Note that the first step of finding the closest prototype of the ith observation, $\min_k \|\mathrm{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{c}_k)\|$, does not need to take the positive weight w_i in (2) into account.

The next issues for the proposed method are the determination of the number of clusters K and the initialization of the clustering algorithm for a given k. Basically, the quality of a cluster can be defined by minimal within-cluster distances and maximal between-cluster distances. Therefore, for the first purpose, we use the approach suggested in [16] and apply three internal cluster indices, namely *Ray-Turi (RT)* [13], *Davies-Bouldin (DB)* [2], and *Davies-Bouldin* (DB*)* [11]. All these indices take both aspects of clustering quality into account: In essence, the clustering error (2), i.e., the sum of the within-cluster distances, to be as small as possible, is divided with the distance between the prototypes (minimum distance for RT and different variants of average distance for DB and DB*), to be as large as possible. When testing a number of possible numbers of prototypes from $k = 2$ into $K_{\max}$, we stop this enlargement when all three cluster indices start to increase.

Concerning the initialization, again partly similarly as in [16], we use a weighted k-means++ algorithm in the initialization of the spatial median based clustering with the weights $\sqrt{w_i}$. A rigorous argument for such an alignment was given in [9] where the relation between variance (weighted k-means) and standard deviation (weighted *k-spatialmedians*) was established. Because of local character, the initialization and the search are repeated $N_s = 10$ times and the solution corresponding to the smallest clustering error in (2) is selected. Furthermore, the weighted k-means++ is applied in the ten initializations with ten different, disjoint data samples (10% of the whole data) that were created using the so-called *Distribution Optimally Balanced, Stratified Folding* as proposed in [12], with the modified implementation given in [7]. Such sampling, by placing a random observation from class j and its $N_s - 1$ nearest class neighbors into different folds, is able to approximate both classwise densities and class frequencies in all the created data samples. Here, we use the 68 country labels as class indicators in stratification.

2.2 Hierarchical application

Because a prototype-based clustering algorithm always works with distances for the whole data, the detection of clusters of different size, especially hierarchically on different scales or levels of abstraction, can be challenging. This is illustrated with the whole PISA data set in Fig. 1, which shows the values of the three cluster indices for $k = 1, \ldots, 68$. For illustration purposes, also the clustering error as defined in (2), denoted as 'Elbow', is provided. All indices have their minimum at $k = 2$ which suggest the division of the PISA data to only two clusters. Note that the geometrical density and low separability of the PISA scale indices might be related to their standardization to have zero mean and unit variance over the OECD countries.

Hierarchical application of the *k-spatialmedians* algorithm was suggested in [17]. The idea is simple: Similarly to the divisive clustering methods, apply the algorithm recursively

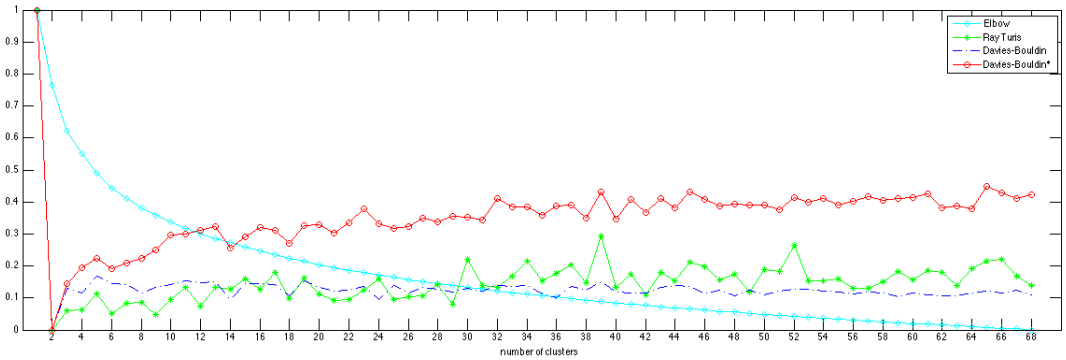


Figure 1: Cluster indices and error slope for the whole sparse PISA data scaled into range $[0, 1]$.

to the cluster data sets that have been determined using the basic approach. For the PISA data here, we realized a recursive search of the *weighted k-spatialmedians* with the depth of three levels, ending up altogether with 2 (level 1), $4 + 4$ (level 2), and $6 + 12 + 10 + 6$ & $2 + 8 + 3 + 6$ clusters (level 3). The wall-clock time for each individual clustering problem was several hours.

3. RESULTS

As discussed in Sec. 1, we use the 15 PISA scale indices that readily summarize most of the students' background as data input for our clustering algorithm. By following the mixture of the partitional/hierarchical clustering approach as described above, we first of all, provide the results of the weighted sparse data clustering algorithm when applied to the whole PISA data (first level). Then, recursively, the results of the algorithm for the newly obtained clusters at the second and third level of refinement are given. For all the clusters at each level, we compute the relative share of students from each country, i.e., the weighted number of students in the cluster in relation to the whole number of 15-year-old students in the country. Moreover, in order to reveal the deviating characteristics of the appearing clusters, we visualize and interpret (i.e., characterize) the cluster prototypes in comparison to the overall behavior of the entire 15-year-old student population in the 68 countries by always subtracting the weighted spatial median of the whole data from the obtained prototypes.

3.1 First Level

Since, as pointed out in Sec. 2.2, all the sparse cluster indices suggest two, we first run our weighted sparse clustering algorithm for $K = 2$. The clustering result on the first level is shown in Fig. 2. The division of these clusters is unambiguous: All scale indices that are associated with high performance in mathematics have a positive value for Cluster 2 and a negative value for Cluster 1. Likewise, those two scale indices that are associated with low performance in mathematics, i.e., the self-responsibility for failing in mathematics (*failmat*) and the anxiety towards mathematics (*anxmat*), show a positive value for Cluster 1 and a negative value for Cluster 2. As can be expected by these profiles, the mean mathematics performance of Cluster 1 is much lower than the mean math performance of Cluster 2 (see Table 1).

When we consider the relative number of students from different countries, we see that every country has students in both clusters. In fact, the distribution of the 15-year-old student population between the two clusters is quite equal in each country. For Cluster 1, the mean percentage of students from a country is 55% while for Cluster 2, the mean is 45%, and both have the standard deviation of 10. In all of the in PISA participating countries and territories, there are higher and lower performing students and it seems that they share the same characteristics. Additionally, the distribution between girls and boys is quite equal, although somewhat in favor of boys: Only 48% of the students in the cluster with the scale indices that are associated with high performance in mathematics are girls. Moreover, the average math score of the boys is in both clusters higher than the average math score of the girls (see Table 1).

Table 1: Characteristics of global/first level clusters

Cluster	**population size** (♀ in %)	**math score** ∅	♀	♂
1	13399687 (52%)	445	442	449
2	11321033 (48%)	468	461	475
all	24720720 (50%)	456	451	461

3.2 Second Level

Following the approach as described above, we run the clustering algorithm again, but this time for each of the two global clusters obtained in the first level separately. According to the same rule given in Sec. 2.1, i.e., stop enlarging k during the search when all the cluster indices are increasing, we get for both of the global clusters $K = 4$ as a number for their subclusters.

3.2.1 *Subclusters of Cluster 1*

Table 2: Characteristics of subclusters of Cluster 1

Cluster	**population size** (♀ in %)	**math score** ∅	♀	♂
1-1	2792046 (56%)	439	438	440
1-2	3873035 (52%)	391	388	394
1-3	3072064 (58%)	466	464	468
1-4	3662542 (45%)	491	489	492

The subclusters of the global Cluster 1 are visualized in Fig. 3 and characterized in Table 2. If we set the threshold

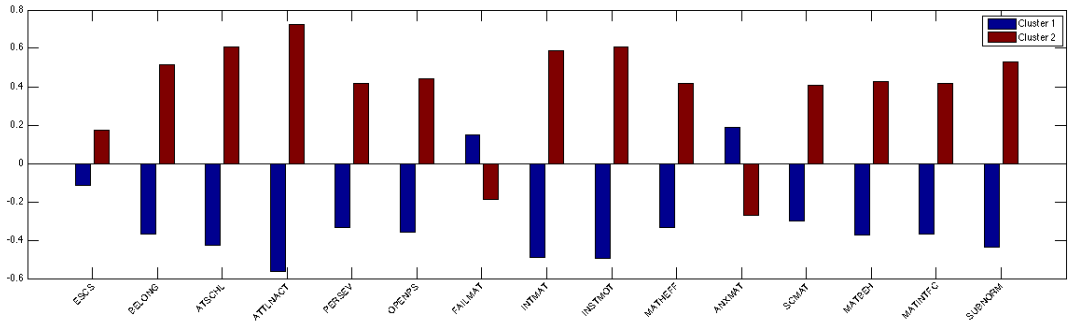


Figure 2: Characterization of the two global clusters.

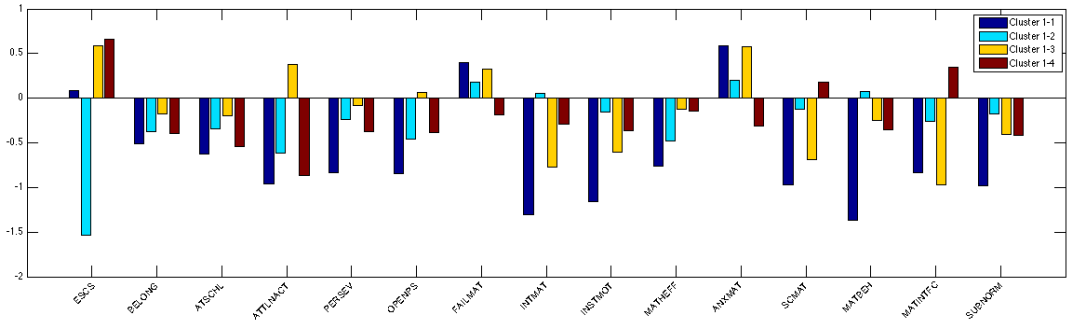


Figure 3: Characterization of the four subclusters of Cluster 1.

of how many students should at least be from one country to 21%, we obtain the following countries for the subclusters: Cluster 1-1 (i.e., subcluster 1 of Cluster 1) contains at most students from East Asia with the exception of China: More than 30% of Japan's 15-year-old student population belongs to this cluster, 26% of Korea's and and 25% of Taiwan's. The remaining students represent a mixture from many different countries which, however, are only represented by less 21% of their 15-year-old student population.

Cluster 1-2 contains almost entirely students from developing countries. Hereby, students from Vietnam form with 49% the majority. Moreover, Indonesia, Thailand (both $> 30\%$) and Brazil, Colombia, Peru, Tunisia, and Turkey (all $> 25\%$) are represented by this cluster. The cluster is, as can be seen from Fig. 3, most notably characterized by a very low economic, social and cultural status (*escs*). That means that the students in this cluster - as a subset of the global Cluster 1 which already represented the more disadvantaged students (see Fig. 2) - are the most disadvantaged.

Cluster 1-3 consists in the majority of students from Eastern Europe: Serbia, Montenegro, Hungary, Slovak Republic (all $> 23\%$) and Romania (almost 22%) constitute the majority. As we can see from Fig. 3, this cluster is the only one in the group of subclusters of the global Cluster 1, that generally was characterized by negative attitudes and perceptions (see Fig. 2), which actually can be distinguished by positive attitudes towards school (*attlnact*). Moreover, it is the cluster with mainly girls in it.

Cluster 1-4 accommodates mainly students from Western and Central Europe. Most of the 15-year-old student population from the Netherlands (39%) are in this cluster, followed by Belgium with 29%, and the Czech Republic with 27%. This cluster is characterized by the highest *escs* among the students of the global Cluster 1. Furthermore, although they have negative values in most of the scale indices, they have a higher self-concept in math, and also much higher intentions to use mathematics later in life in comparison with their peers.

3.2.2 *Subclusters of global Cluster 2*

Table 3: Characteristics of subclusters of Cluster 2

Cluster	**population size** (♀ in %)	**math score** ∅	♀	♂
2-1	3127958 (43%)	526	523	528
2-2	2739481 (54%)	457	457	458
2-3	3521092 (50%)	400	397	403
2-4	1932502 (44%)	515	506	523

The subclusters of the global Cluster 2 are characterized in Fig. 4 and summarized in Table 3. Again, we search for clusters that mostly deviate from the others. Cluster 2-1 is such a cluster: The students in this cluster have the highest average math score (see Table 3), the highest intentions to pursue a mathematics related career but a sense of belonging to school (*belong*) and subjective norms in mathematics (*subnorm*) that are only about the same as the average of the whole 15-year-old student population (see Fig. 4). The subjective norms in mathematics measure how people important to the students, such as their friends and parents, view mathematics. In the global Cluster 2, those students

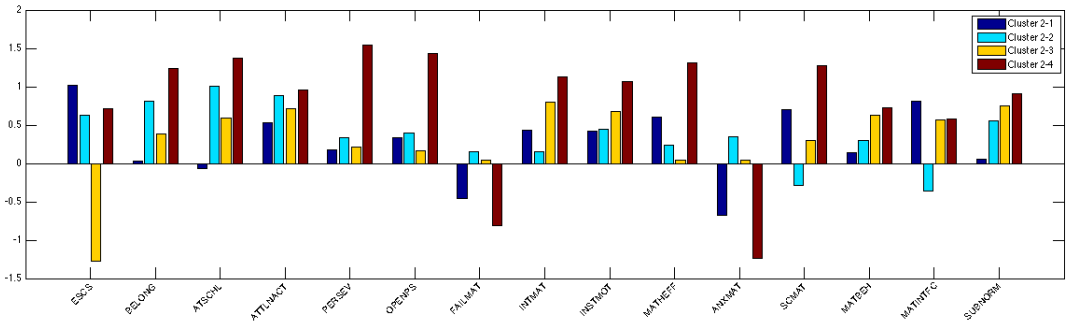


Figure 4: Characterization of subclusters of Cluster 2.

who had high positive values in the other scale indices associated with high performance in mathematics, also thought that their friends and family view mathematics as important (their *subnorm* value is very high, see Fig. 2). Students in this cluster, however, seem not to be influenced or affected by what people close to them think. It appears to be a rather strong cluster that also has the highest percentage of boys in it. For this cluster, we again compute the relative number of students from each country. And indeed, it shows a very clear country-profile. The highest percentage of students come from the English-speaking and Nordic countries: Denmark (more than 30%), Iceland and Sweden (both $> 26\%$) have the highest percentages of their 15-year-old student population in this cluster. Followed by the two highest performing districts in the USA, namely Connecticut and Massachusetts, with both more than 25%. Besides these countries and territories, the cluster has also a high share of students from Norway, Finland, Great Britain, Australia, and Canada (almost 22% or more). Additionally, the USA has with more than 21% still a relatively high share of students in this cluster. According to the Hofstede Model (see Sec. 1), all of these countries are characterized by high individualism.

Also Cluster 2-3 shows an explicit country profile: 36% of the 15-year-old student population from India are in this cluster. Moreover, the cluster consists of students from Peru and Thailand (both 30%), Turkey (27%) and Vietnam (26%). Altogether, we find here the most disadvantaged students (indicated by the very negative *escs*) among the subgroups of the global Cluster 2 and the largest share of students come from the developing countries. However, these students have very positive attitudes towards education and show relatively high values in all scale indices that are associated with high performance in mathematics.

To this end, Cluster 2-2 and Cluster 2-4 have less obvious country affiliations. Cluster 2-2 can at best be described as containing mostly countries with Islamic culture. Most of the students are from the United Arab Emirates and Albania (both 21%), Kazakhstan and Jordan (both 19%). According to the Hofstede Model, these countries are similar in that way that they all show very high power distance. Cluster 2-4 has with 25% the highest share of students also from Kazakhstan, but the remaining countries in this cluster (all have less than 17% of their 15-year-old students population in it) are widely mixed.

Altogether, among the clusters at the second level, Cluster 2-1 appears to be the most interesting one, i.e., the most distinct group with the clearest country profiles.

3.3 Third Level

Recursively, we repeat the same approach on the next level, i.e., for the subclusters of the eight clusters identified in Sec. 3.2. For all the new subclusters, the best number of clusters as determined by the cluster indices are as follows: 6, 12, 10, and 6 for the four subclusters of the first global cluster, and 2, 8, 3, and 6 for the four subclusters of the second global cluster. This means that we have 53 different clusters on this level - almost as many as different countries/territories in the whole PISA 2012 data. If our hypothesis is true, we should be able to find clusters that clearly contain more students from certain countries. Exactly as in Sec. 3.2, we first of all compute the basic facts of each cluster. Note, however, that the deeper we go in the hierarchy the more clusters we encounter and the more difficult it becomes to define clear rules and thresholds to distinguish significant characterizations of clusters.

3.3.1 Subclusters of Cluster 1-3

Table 4: Characteristics of subclusters of Cluster 1-3

Cluster	**population size** (♀ in %)	**math score** ∅	♀	♂
1-3-1	335240 (61%)	493	492	495
1-3-2	262779 (48%)	539	540	538
1-3-3	368591 (51%)	461	460	462
1-3-4	273629 (66%)	492	491	492
1-3-5	359721(56%)	427	428	426
1-3-6	275513 (63%)	437	436	438
1-3-7	264017 (63%)	443	441	447
1-3-8	318607 (63%)	460	457	464
1-3-9	216704 (60%)	421	418	424
1-3-10	397263 (56%)	481	482	480

The first interesting cluster appears in the 1-3 group. Cluster 1-3-8 accommodates mainly students from South West Europe: Austria, Liechtenstein, Spain, France, and Italy. According to the Hofstede Model, all of these countries are depicted by high avoidance of uncertainty.

3.3.2 Subclusters of Cluster 1-4

The characterization of the subclusters in the 1-4 group are provided in Fig. 6, and summarized in Table 5. Also here,

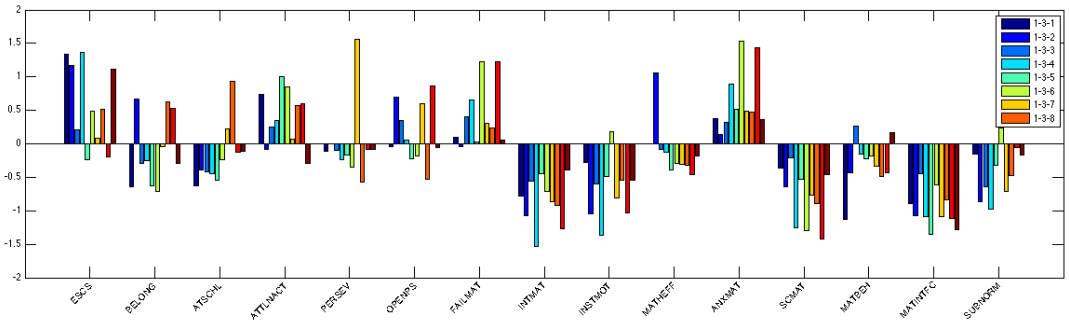


Figure 5: Characterization of subclusters of Cluster 1-3.

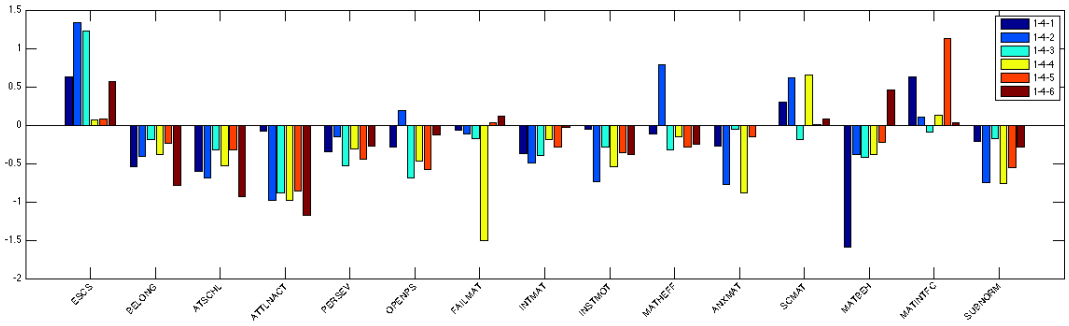


Figure 6: Characterization of the subclusters of Cluster 1-4.

Table 5: Characteristics of subclusters of Cluster 1-4

Cluster	population size (♀ in %)	math score ∅	♀	♂
1-4-1	485599 (48%)	481	480	482
1-4-2	520763 (38%)	556	558	555
1-4-3	771799 (53%)	494	494	495
1-4-4	489528 (43%)	497	491	501
1-4-5	754515 (48%)	470	467	473
1-4-6	640338 (38%)	461	465	458’

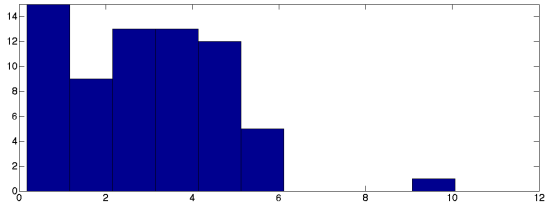


Figure 7: Histogram of the distribution of countries from the students in Cluster 1-4-2.

we are searching for explicit country clusters. This search is realized by looking at the histograms and identifying those clusters that for some countries have a considerably higher share of their 15-year-old student population in it than for the remaining countries. The histogram in Fig. 7 shows one example of this for Cluster 1-4-2: In this cluster, the portion of students in it deviates significantly from the others for exactly one country with 10% of its 15-year-old student population. This country is the Netherlands. For all other countries, the share of their 15-year-old student population in this cluster is less than 6% (see Fig. 7). As can be seen from Fig. 6, this ‘Netherlands Cluster’ is characterized by having the highest math self-efficacy amongst its group.

Cluster 1-4-1 is again a mixture of Nordic and English-speaking countries. The highest share of students in this cluster come from the United Kingdom, Ireland, Norway, New Zealand, and Sweden. As these two country profiles were already detected to be in the same cluster on the higher cluster level (see Sec. 3.2.1), it really seems that students from these countries share many similar characteristics.

Cluster 1-4-4 has the highest share of East Asian countries including two of the three districts of China that participated in PISA 2012. Most of the students in this cluster come from Japan, followed by Taiwan, Macao-China and Hong Kong-China. One of the most distinct feature of this cluster is, as can be seen from Fig. 6, the high self-concept in mathematics (*scmat*). According to the Hofstede Model (see Sec. 1), all of these countries show high pragmatism.

3.3.3 Subclusters of Cluster 2-1

Table 6: Characteristics of subclusters of Cluster 2-1

Cluster	population size (♀ in %)	math score ∅	♀	♂
2-1-1	1346930 (40%)	562	557	566
2-1-2	1781028 (45%)	498	500	497

From Sec. 3.2, we concluded that Cluster 2-1 was the most interesting one. Moreover, Cluster 2-1 was the cluster that had the highest share of two country profiles in it: On the one hand, the English-speaking countries, and, on the other

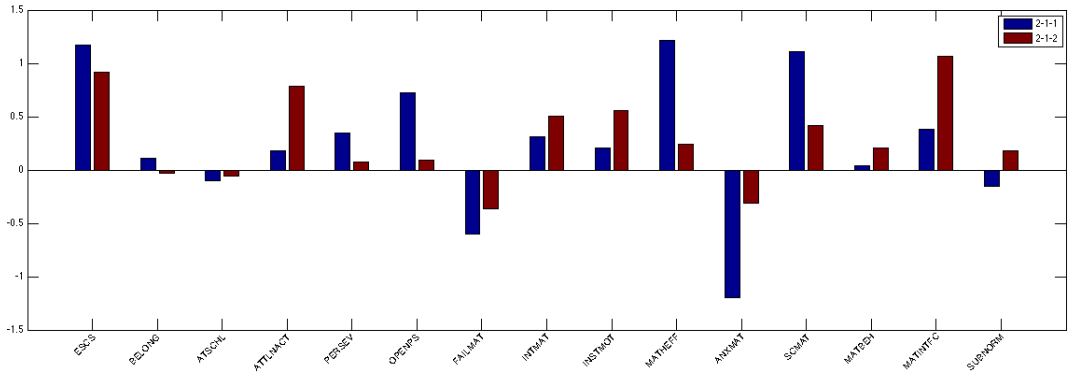


Figure 8: Characterization of subclusters of Cluster 2-1.

hand, the Nordic countries. Interestingly, the cluster indices also suggest to divide this cluster into two further countries. However, when we look again at those countries that have the highest percentages of their 15-year-old students, the two clusters still contain mostly students from both country profiles. For example, 15% of the Danish 15-year-old student population are in Cluster 2-1-1, and 14% are in Cluster 2-1-2. Similarly, 14% of the 15-year-old student population from Connecticut are in Cluster 2-1-1, and 11% in Cluster 2-1-2. Apparently, this cluster does not divide any further between Nordic and English-speaking countries. It only divides the high-performing students from these countries into two types: On the one hand, the type that has a very high self-efficacy (*matheff*) as well as self-concept (*scmat*) in math, i.e., the students that have a very high belief in their own ability, and, on the other hand, the type that has very high intentions to pursue a math related career (*matintfc*).

However, also a new clear group of countries appears. Cluster 2-1-1 has a very high share of German-speaking countries in it: More than 12% of Germany's and Switzerland's 15-year-old student population, and 10% of Austria's can be found in this cluster. None of these countries appear in the sibling Cluster 2-1-2 when the threshold is set to 9%. It seems that high-performing German-speaking students feel very confident in solving mathematical tasks but only show a moderate positive value in the intentions to use mathematics later in life, a characteristic that one would associate the most with the traditional functional German stereotype (see Sec. 1) that is expected to attach great importance to utilitarianism [4]. According to the Hofstede Model, all of these three German-speaking countries are considered to be highly masculine.

3.3.4 *Subclusters of Cluster 2-4*

Table 7: Characteristics of subclusters of Cluster 2-4

Cluster	**population size** (♀ in %)	**math score** ∅	♀	♂
2-4-1	186107 (37%)	533	528	536
2-4-2	430729 (40%)	582	575	588
2-4-3	261838 (45%)	440	436	443
2-4-4	378120 (50%)	477	468	486
2-4-5	430105 (47%)	520	519	521
2-4-6	245603 (40%)	516	500	526

The subclusters of Cluster 2-4 are summarized in Table 7 and characterized in Fig. 9. The clearest country profile among this group is 2-4-6: It consists to the highest share of students from high-performing Asian countries: Shanghai-China and Singapore. As we can see from Fig. 9, similarly to Cluster 1-4-4 (see Sec. 3.3.2) that also contained a high share of East Asian students, this cluster is characterized as well by a high self-concept in mathematics (*scmat*). The students in this cluster believe that mathematics is one of their best subjects, and that they understand even the most difficult work. Furthermore, as already found for Cluster 1-4-4, also for this cluster the main countries show high pragmatism according to the Hofstede Model.

4. CONCLUSIONS

In this article, we have introduced a clustering approach that has both partitional and hierarchical components in it. Moreover, the algorithm takes weights, aligning a sample with its population into account and is suitable for large data sets in which many missing values are present.

The hypothesis in our study was that the different clusters determined by the algorithm, when all students with their attitudes and behaviors towards education are given as input, could be explained by the country of the students in particular clusters. Our overall results on the first level showed that in each cluster students from all countries exist and that the actual test performance (as well as a simple division in positive and negative attitudes towards education) explain the clusters much better than the country from which the students in the particular cluster come from.

However, on the next two levels many clusters were detected that obviously had a much higher share of students from certain countries. For example, an Eastern Europe, a German-speaking, an East Asia, and a developing countries cluster were identified. On the second level, also a very clear cluster that consisted to a high portion of Nordic and English-speaking countries appeared. This cluster did not split further on the next level to fully separate these two distinct country profiles. Instead, the cluster was divided into two student types, of which both the Nordic as well as the English-speaking countries seem to have an almost equal share of their students from.

Summing up, we conclude that groups of similar countries,

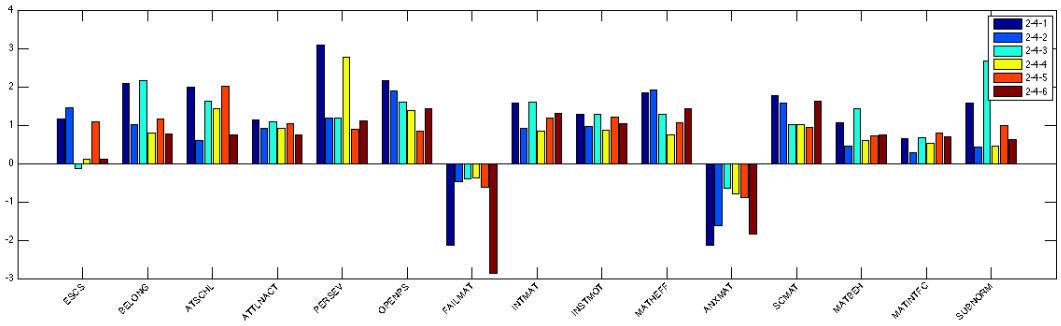


Figure 9: Characterization of subclusters of Cluster 2-4.

e.g., by means of geographical location, culture, stage of development, and dimensions according to the Hofstede Model, can be found by clustering PISA scale indices but the actual country stereotypes exist only to a very marginal extent. However, in a further work the rules how to find relevant clusters could be improved and more variables than the 15 scale indices utilized here could be included to the algorithm. The PISA scale indices are linked to math performance and in every country there are higher and lower performing students who share similar overall characteristics. Nevertheless, we think that the overall results presented here show a very promising behavior already, and we expect that the resulting clusters of our algorithm could be explained even clearer by the country of the students if additional information such as the students' temperament would be available for the clustering algorithm.

References

[1] S. Äyrämö. *Knowledge Mining Using Robust Clustering*, volume 63 of *Jyväskylä Studies in Computing*. University of Jyväskylä, 2006.

[2] D. L. Davies and D. W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.

[3] S. Harrell. Why do the Chinese work so hard? Reflections on an entrepreneurial ethic. *Modern China*, pages 203–226, 1985.

[4] M. F. Herz and A. Diamantopoulos. Activation of country stereotypes: automaticity, consonance, and impact. *Journal of the Academy of Marketing Science*, 41(4):400–417, 2013.

[5] T. P. Hettmansperger and J. W. McKean. *Robust nonparametric statistical methods*. Edward Arnold, London, 1998.

[6] G. Hofstede. Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture*, 2(1):8, 2011.

[7] T. Kärkkäinen. On cross-validation for MLP model evaluation. In *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science (8621), pages 291–300. Springer-Verlag, 2014.

[8] T. Kärkkäinen and E. Heikkola. Robust formulations for training multilayer perceptrons. *Neural Computation*, 16:837–862, 2004.

[9] T. Kärkkäinen and M. Saarela. Robust principal component analysis of data with missing values. *To appear in the Proceedings of the 11th International Conference on Machine Learning and Data Mining MLDM*, 2015.

[10] T. Kärkkäinen and J. Toivanen. Building blocks for odd-even multigrid with applications to reduced systems. *Journal of Computational and Applied Mathematics*, 131:15–33, 2001.

[11] M. Kim and R. Ramakrishna. New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353–2363, 2005.

[12] J. Moreno-Torres, J. Sáez, and F. Herrera. Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1304–1312, 2012.

[13] S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, pages 137–143, 1999.

[14] M. Saarela and T. Kärkkäinen. Discovering Gender-Specific Knowledge from Finnish Basic Education using PISA Scale Indices. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 60–68, 2014.

[15] M. Saarela and T. Kärkkäinen. Analysing Student Performance using Sparse Data of Core Bachelor Courses. *JEDM-Journal of Educational Data Mining*, 7(1):3–32, 2015.

[16] M. Saarela and T. Kärkkäinen. Weighted clustering of sparse educational data. *To appear in 23rd Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

[17] P. Wartiainen and T. Kärkkäinen. Hierarchical, prototype-based clustering of multiple time series with missing values. *To appear in 23rd Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

Student Privacy and Educational Data Mining: Perspectives from Industry

Jennifer Sabourin Lucy Kosturko Clare FitzGerald Scott McQuiggan
Curriculum Pathways
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC, USA
1.919.677.8000
{Jennifer.Sabourin, Lucy.Kosturko, Clare.FitzGerald, Scott.McQuiggan}@sas.com

ABSTRACT
While the field of educational data mining (EDM) has generated many innovations for improving educational software and student learning, the mining of student data has recently come under a great deal of scrutiny. Many stakeholder groups, including public officials, media outlets, and parents, have voiced concern over the privacy of student data and their efforts have garnered national attention. The momentum behind and scrutiny of student privacy has made it increasingly difficult for EDM applications to transition from academia to industry. Based on experience as academic researchers transitioning into industry, we present three primary areas of concern related to student privacy in practice: policy, corporate social responsibility, and public opinion. Our discussion will describe the key challenges faced within these categories, strategies for overcoming them, and ways in which the academic EDM community can support the adoption of innovative technologies in large-scale production.

Keywords
Student privacy, student data, policy

1. INTRODUCTION
Educational data mining (EDM) is chiefly defined by the application of sophisticated data mining techniques to solving problems in education [1]. A powerful tool, EDM has been successfully incorporated into applications that optimize student learning in both research and commercial products. EDM's proven effectiveness has led many—from the U.S. government to individual teachers—to recognize the ability of student data in guiding education and to support the development and use of these technologies in schools. Consequently, applications utilizing EDM technologies have become more prevalent in school systems [2], [3].

However, the increase in EDM usage has raised public awareness of how much data is being collected about students. The applications and companies that collect and use student data are coming under scrutiny, as parents, advocates, and public officials grow concerned over student privacy. A recent cascade of events has focused attention on privacy concerns [4]. For example, there has been a rise in high-profile attacks on consumer data from online retailers and financial institutions. Large, well-trusted institutions have been targeted for using student data in undesirable ways [5]. Promising companies driven by student data have been brought down by public opinion with no evidence of wrong-doing. Calls for stricter policy from privacy advocates have led to more than 100 bills being introduced in U.S. state legislatures to address issues of student privacy in 2014 [4]. In response, the White House has announced plans for federal legislation modeled after state policies [6].

Negative media attention and increased legislation threaten to stifle EDM, particularly in commercial settings. Public opinion may make organizations wary to invest in and use EDM techniques while legislation could make it more difficult to collect and use student data in effective ways. We believe it is an incredibly important time for the EDM community to be aware of the challenges being faced in industry. The rise of concern over student privacy has strong implications for how new EDM approaches can be integrated into wide-reaching applications as well as the amount of funding available to public and private entities wishing to innovate in this space.

These issues are receiving rapidly increasing attention and driving action at the national level. It is critical that the discussions around these issues include experts from the EDM community. This paper discusses the issues and implications faced by commercial applications of educational data mining because of recent focus on student privacy. In this paper, we discuss the role of policy, corporate social responsibility, and public opinion in framing the work of and challenges to industry. We discuss strategies for overcoming these challenges and present opportunities for the EDM community to address rising concerns.

2. EDM AND INDUSTRY
The profile of the EDM community has risen in the past decade—in research, commercial products, public attention—bolstered by three related shifts. First, educational technology has been more widely adopted. School systems are investing in laptops, mobile devices and other technologies in favor of static textbooks. These technologies offer opportunities for data collection that did not exist before. Student records are also increasingly digitized including test scores, attendance records, and bus schedules. These digitized records have generated a wealth of longitudinal data that was previously difficult and expensive to collect [7].

Second, there has been a dramatic rise in computational power and storage capacities. This storage allows for the collection and housing of large amounts of data, even data that is not presently known to be useful. The increased computational power has generated sophisticated algorithms that can mine large corpora of data to identify connections that would previously be impossible [8] and has even created the possibility for robust decision engines to operate in real time learning systems.

Finally, public officials and industry experts are starting to recognize the power of educational data mining [9]. Government funding opportunities for data-driven education solutions are on the

rise, and reports estimate that educational data mining has the potential to provide meaningful economic impact worldwide [10].

There are many areas of EDM research, each with unique applications to industry. At the individual level, data on student behavior, from mouse clicks to eye tracking, provide insight on how students interact with educational technology. For example, EDM has produced models of help abuse [11], attention to hints [12], and conversational dynamics in online forums [13]. These insights and techniques can help commercial educational technology providers design better applications that support positive interactions with students while being user-friendly.

Another key area of research at the individual level is assessment. EDM applications have been used to identify student mastery as well as knowledge gaps. Frequently, these models are based on student performance on relevant tasks but can go beyond measuring what students did correctly and incorrectly by modeling underlying knowledge [14]. Some assessments are cleverly hidden, called "stealth assessment," in games or other non-threatening applications [15]. These systems develop robust models of student knowledge while avoiding the negative effects associated with test performance; in fact, students may not even knowing they are being tested. These techniques have important implications for educational technologies, ranging from the design of new systems that can revolutionize the way assessment is done in formal learning environments, to technologies that can identify gaps in student knowledge and recommend resources to help fill them.

EDM technologies have also driven personalized learning beyond tailoring instruction to what students know, but also to how they learn based on needs and preferences. Systems can identify commonly used strategies by students and select which are most effective, for particular individuals, under specific circumstances [16]. EDM techniques have also supported technologies that guide students towards learning how to regulate their own learning, by helping them to recognize and overcome weaknesses in their current approaches [17]. These techniques are critical in creating applications that use the most effective techniques and support personalized learning.

Finally, EDM research has examined mining data at higher levels, including schools and districts, for a variety of purposes such as exploring college readiness [18], identifying the best teachers [19], or driving district spending [7]. Commercial products are commonly used to house this level of data and communicate findings to necessary stakeholders. Data mining on this organizational or even regional level has allowed for the development of early warning systems to predict student drop-out before it happens as well as identify holes in district-level education [7].

In essence, "educational data mining and learning analytics have the potential to make visible data that have heretofore gone unseen, unnoticed, and, therefore, unactionable" [9]. The approaches outlined in this section offer significant promise in helping to improve education delivery and outcomes, but their success is contingent on the collection, storage, and use of large amounts of quality student data. Companies who wish to collect and use student data must operate under increased public and governmental scrutiny, which can, and has, created barriers to the use of EDM in industry.

3. STUDENT PRIVACY

Privacy is chiefly a question of access. Unlike anonymity or confidentiality, peoples' interest in privacy is about controlling the access of others to themselves [20]. How to safeguard a child's privacy is a particularly complex question because of their vulnerability. Children are incapable of "protecting their own interests through negotiation for informed consent" because they are likely to misunderstand risks or be coerced into participating [20].

This need to protect has led to the formation of student privacy advocacy groups and driven the adoption of legislation. The restrictions required to comply with this legislation and maintain good public opinion have a significant impact on the adoption of data-based solutions in education.

3.1 Policy

In the U.S., we have established privacy protections for children by asking for consent from parents or guardians and implementing policies which hold organizations, both public and private, accountable for obtaining consent when collecting, storing or disclosing data, and ensuring proper usage. There are two federal acts that address children's privacy directly: the Federal Education Rights and Privacy Act (FERPA), and the Children's Online Privacy Protection Act (COPPA).

3.1.1 Federal Education Rights and Privacy Act

Before the enactment of the Federal Education Rights and Privacy Act (FERPA) in 1974, parents and students had little access to education records. Meanwhile, that same information was widely available to outside authorities without requiring the consent of parents or students [21]. FERPA applies to any school receiving federal funds and levies financial penalties for not following it. While complying with FERPA is a local responsibility [22], the way it defines education records and regulates third party access to them matters to private companies.

According to FERPA, education records contain information on student background, academic performance, grades, standardized test results, psychological evaluations, disability reports, and anecdotal remarks from teachers or school authorities regarding academic performance or student behavior (FERPA, 1974, 20 U.S.C § 1232g(a)(1)(D)(3). Generally, schools looking to disclose information contained in these records must have written permission from a parent or eligible student, an individual who is 18 or attending post-secondary school. Education record information is only shared with a third party on the assurance that that third party will not allow further outside access to requested information without additional written parental consent (FERPA, 1974, 20 U.S.C § 1232g (b)(4)(B)). Some activities, however, do not require written consent. Under FERPA, third parties, including private companies, may use information within education records for official or contracted evaluation, audit, and compliance activities without parental or student consent but are barred from using that data for marketing [23].

FERPA is not without controversy. Some have argued that schools improperly apply FERPA in order to protect information that does not fall under its definition of an education record and that such denials of disclosure are in violation of state open record laws [24]. Others voice concern over contracted service providers' use of data not covered by FERPA citing that the content of emails housed in cloud services, data from identification cards, or data collected by schools to outsource a service could, depending on the contract, be used or sold for marketing purposes [23].

3.1.2 Children's Online Privacy Protection Act

While FERPA affects private interests, the Children's Online Privacy Protection Act (COPPA) speaks more directly to

operations, particularly to online service providers that have direct or actual knowledge of users under 13 and collect information online. Made effective in 2000, COPPA "requires web hosts and content providers to seek parental consent to store data about children under age 13" [25]. To be fully compliant, parents must be given the opportunity to review terms of service and privacy policies of each commercial website where their child's information may be stored. Parental consent is required before any information can be collected, and parents can retract this permission and request all data be deleted at any time. Technology providers are required to disclose what data is being collected about children and what it is being used for. They are also expected to provide reasonable measures of security and discard of data once it is no longer needed. [25], [26]. Overall, COPPA seeks to encourage responsible business practice and reduce "imprudent disclosures of personal information by children" [27].

COPPA, too, has fallen under criticism. It is difficult to enforce and there many ways in which companies can comply with the "letter of the law" without truly protecting student privacy. COPPA has also been criticized for not reflecting the changes in online technologies accessed by children. In an effort to stay current with technological advancement, COPPA underwent revisions in 2013 to "address changes in the way children use and access the Internet, including the increased use of mobile devices and social networking" [28] by widening the definition of what constitutes children's personal information to include cookies, geolocation, photos, videos, and audio recordings [28]. These updates bolstering safeguards for student data appear further scaffolded by actions from the White House.

3.1.3 Student Digital Privacy

Driven by concerns over the efficacy of national policies, state legislators have seen the introduction of a large number of policies aimed at protecting student data [4], [29]. New national legislation may also be on the horizon for protecting student privacy [30]. The proposed Student Digital Privacy Act, modeled after a California statute, prohibits companies from selling student data to third parties except for educational purposes [6]. While it is unclear when, or if, this legislation will be enacted, it has already drawn criticism. Parents and privacy advocates fear it is too lenient while industry experts warn that increased legislation may limit development of important educational solutions [31].

These industry experts point to the voluntary Student Privacy Pledge (http://studentprivacypledge.org/) as a means to achieve better management of student data without federal legislation [32]. At the time of writing, 108 companies have chosen to sign the pledge, vowing that they will not sell student data or use data for targeted advertisement, and will maintain transparency about how data is being collected and used. This pledge is an indication that commercial education technology providers are taking steps towards the corporate social responsibility that will garner respect among users and privacy advocates.

3.1.4 Student Privacy: International Perspectives

The United States has relied on a piecemeal approach to regulating privacy where legislation is sector driven and may be enacted at state and/or federal levels [33]. Conversely, the European Union enacted a comprehensive set of regulations in the Data Protection Directive under which student privacy issues are largely subsumed. This set of regulations requires unambiguous consent of individuals before collecting or processing personal data as well as a prohibition on collecting sensitive information with few exceptions [34].

Canadian national privacy legislation is stipulated in the Personal Information Protection and Electronic Documents Act which, like COPPA, is focused on how commercial entities use personal information, as well as the Privacy Act which limits the collection, use, and disclosure of personal information by federal government entities. Meanwhile, similar to United States, Canadian provinces follow their own patchwork of student specific legislation. Ontario, for instance, follows the Education Act, the Municipal Freedom of Information and Protection of Privacy Act as well as the Personal Health Information Protection Act. The Canadian system is less comprehensive than the EU, but is perhaps more effective in safeguarding student interests than the US due to an "all-encompassing and prescriptive nature" [34].

3.2 Corporate Social Responsibility

Corporate social responsibility refers to companies taking an active part ensuring they have a positive impact on social welfare. In the case of privacy, this means working to truly protect student data and collect and use it responsibly. Design weaknesses and enforcement shortcomings in student privacy legislation can often allow companies to appear more responsible than they are. Organizations can legally comply, a potentially cumbersome process on its own, but do little to actually ensure best practices are being followed and student interests are protected.

This is a significant issue in markets of educational technologies designed for children under the age of 13, the population protected by COPPA. True compliance with the intents behind COPPA can be "both overwhelming and prohibitive" [35] which privacy scholar, Danah Boyd, believes has led to an apprehension to target users under thirteen. Avoiding the issue is often seen as "easier and more cost effective than attempting to tackle COPPA compliance." [35]

Currently there are many websites, online services, and mobile apps that are widely used in classroom settings including those classrooms with younger students. For example, Google Apps for Education reportedly serves an estimated 40 million students, teachers, and administrators. Similarly, over 47 million teachers have accounts with Edmodo, the "world's largest K-12 social learning community". Education technology is estimated to be an 8 billion dollar industry [30] and technology providers are often trying to find their niche while maintaining competitive advantage. Issues arise when creating a product that will be useful to education, ensuring that student data is collected and managed responsibly, and managing profit and competition are at conflict with one another. This balance of constraints is one of the strongest challenges faced by companies seeking to gather and use educational data responsibly.

3.2.1 Supporting Shared-Device Settings

Classroom constraints make the educational market particularly unique. While 1:1 schools (1 device per student) and Bring Your Own Device (BYOD) integrations are on the rise, many schools reflect a shared-device model (e.g., classroom sets, device carts). In order to achieve personalized learning in this setting, individual accounts are often necessary. Yet individual accounts raise several issues.

The first is that secure account authentication can be troublesome. Expecting students, especially younger students, to remember their login credentials is unreasonable in many cases. Keeping up with login information is particularly challenging when classrooms attempt to take advantage of multiple systems each requiring their own unique username and password. In fact, a report by the National School Board Association notes "password reuse due to

lax controls (i.e., password written on a sticky note)" as a particular concern for using online educational services [36]. Some systems utilize password pictures or avatars for younger populations, which could be a viable option depending on the type of data; however, when sensitive data such as images, video, and performance evaluations are often protected behind account logins, it is important to enable users to securely protect their data.

Furthermore, for those companies without any interest in storing student data on servers, shared-device settings can unintentionally force this responsibility. In a 1:1 environment, user data can simply be stored on students' devices as there is little concern over other individuals gaining access to the data; thus, eliminating the need to device solutions for complying to privacy legislation and avoiding security breaches. Appealing to shared-device environments, on the other hand, necessitates such measures including cloud storage, a solution known to concern parents [37]. Moreover, when schools rely on online educational resources and mobile apps that utilize cloud storage, they often relinquish control of that student data, which is particularly alarming given the fact that FERPA "generally requires districts to have direct control of student information when disclosed to third-party service providers" [23]. A recent report by Fordham Law School on the issue of student privacy and cloud computing found "school district cloud service agreements generally do not provide for data security and even allow vendors to retain student information in perpetuity with alarming frequency" [23]. The report goes on to point out that "fewer than 25% of the agreements [pulled from a national sample and reviewed by the committee] specify the purpose for disclosures of student information, fewer than 7% of the contracts restrict the sale or marketing of student information by vendors, and many agreements allow vendors to change the terms without notice." In sum, supporting ubiquitous student access through cloud computing necessitates a great deal of legal accommodations.

3.2.2 Consent

The process for simply creating an account can be cumbersome and time-consuming for two primary reasons: 1) companies cannot collect personal information from students under thirteen without parental consent, and 2) students under 18 cannot legally agree to the Terms of Service agreement accompanying many registration processes. In some cases, schools obtain a blanket agreement from parents at the beginning of the year allowing instructors to create accounts for students. Although, if teachers do not have legal consent from parents to create accounts on their students' behalf, having to wait for parental approval can easily derail an entire lesson quickly making the resource obsolete to the instructor.

Unfortunately, many companies find "restricting" users, even audiences for which the product is intended, streamlines the registration process by avoiding parental consent. Susan Fox of the Walt Disney Company articulates this concern by stating, "Operators are keenly aware that consumers will quickly move on if websites are slow to load, functionality is delayed, or registration-type processes stand between users and their content." [38] Furthermore, because virtual age verification is difficult and easily bypassed, compliance can still be met by adding statements such as "we do not knowingly collect data" from persons under thirteen in privacy policies. As a result, sidestepping the intentions of COPPA makes it difficult for other companies to remain competitive and "discourage[s] startups from innovating for the under-thirteen market" [38].

3.2.3 Disclosure

Parental consent and disclosure are two of the major tenants of COPPA compliance. Responsible adherence suggests that companies are forthcoming with information and present details clearly to parents when asking consent. However, this can be troublesome and may serve to harm parental opinions of an application rather than help. For example, there is concern that anything requiring parental permission (e.g., PG-13 or R-rated movies) is somehow objectionable. This misconception stems from the fact that "parents and youth believe that age requirements are designed to protect their safety, rather than their privacy." [39] As a result, companies attempting to be compliant may be inadvertently penalized because of public opinion.

Privacy policies are another form of disclosure that may be open to misinterpretation. Regulated by the FTC, privacy policies require companies to be upfront about the collection and use of user data. There is, however, much debate about their effectiveness. In a recent survey, over half of interviewed online Americans agreed with the statement, "When a company posts a privacy policy, it ensures that the company keeps confidential all the information it collects on users" and even fewer users read—or, in the case of these younger populations, can read and comprehend—them [40]. Others have proposed alternative solutions that more clearly convey the purposes of data collection [41] yet truly articulating the intricacies of EDM and personalized learning environments will take proofs of concept and time.

3.3 Public Opinion

One of the largest drivers behind the focus on privacy of student data is the vocal concern of parents and stakeholders in the media. The issue has been gaining a great deal of attention and has already had serious impacts on the landscape of educational technology providers.

Perhaps one of the best examples of the power of backlash from parents and media is the demise of a well-funded nonprofit company based entirely on the promise of educational data mining [5]. Though it was widely supported by districts, industry experts, and funding agencies, its efforts were undermined by parental protests and media frenzy. The company did not respond to rising concerns and failed to staunch fears over data misuse and protection. Though there was no evidence of any wrong-doing on the part of the company, parents and privacy advocates protested that the risk was too great. As the protest grew larger and more vocal districts began withdrawing participation in early 2014.

While anecdotal, this example demonstrates the need for industries relying on student data to get ahead of the rising panic by demonstrating value (i.e. driving innovation and/or supporting student learning). While EDM has its proponents [2], [9], their beliefs do not propagate to the general public. Parents and privacy advocates do not believe the benefits to be gained by educational technologies driven by student data outweigh the risks. The top concerns for these individuals are varied, as are their levels of awareness with various issues. Commonly discussed areas of concern with regards to student data include marketing, security, decision-making, and the "unknown".

3.3.1 Marketing

A primary purpose behind existing and proposed legislation is to limit the use of children's data to drive targeted advertisements [42]. It is, therefore, unsurprising that this is one of the top concerns of parents and school officials. However, much of this legislation and parental concern stems from children's interactions with non-educational sites and technologies. In this case, it makes sense to

limit targeted advertising of toys, food items, and other commercial goods, especially when considering findings that children are mostly unable to distinguish advertisement from regular content [43].

However, it is not clear that this protection is warranted in educational contexts. Much of the "advertisement" promoted by the EDM community centers around identifying gaps in a student's understanding and surfacing the most effective and engaging ways to fill those gaps. These advertisements have strong potential to benefit students, but some parents and other privacy advocates are only able to see that their children are being exploited for profit [2][3].

3.3.2 Decision-Making

Several EDM technologies provide a promise to support data-driven decisions about how best to help students learn. This is seen regularly in tools that select problem sets, feedback, or lesson plans based on students' prior interactions [44]. Data may also be presented to educators or administrators making decisions about whether a student needs additional attention or if they are college-ready [18]. These types of decisions start drawing parental concern. While parents understand (though they may not agree with) data from high stakes examinations being used to drive decisions about their children's education, data from private learning technologies is more unclear. Parents fear that undisclosed "stealth assessments" could negatively impact their children's future – from academics through the work force [42].

3.3.3 Security.

In addition to concerns over what companies may do with the data they collect, many parents are also fearful over what may happen if that data enters the wrong hands. The news is rife with incidents of data breeches with individual financial and other personal data being accessed by malicious parties. Parents concerns over student data security is certainly valid, though experts think it unlikely that this type of data would draw attack as it is less obviously lucrative when compared with financial and other personal records [2].

Existing legislation does put restrictions on the collection and storage of personally identifiable information (PII) of minors and responsible companies do strive to ensure anonymization of data. However, the rapid increase in the quantity of data collected and the sophistication of data mining procedures increase the likelihood that data that does not seem like PII on the surface could be combined to identify individuals [8].

3.3.4 The "Unknown".

Finally, many fears from parents and the media cannot be vocalized. There is something unsettling about the quantity of data being collected, stored, and mined about children, even if there is no real threat to safety or happiness. Much of this fear stems from the lack of transparency that surrounds the issues. Companies want to keep practices secret to avoid giving competitors an advantage. Privacy policies are often vague and uninformative to reduce the risk of drawing criticism or lawsuits. This is especially a concern as media tensions and attacks rise. Parents know that large quantities of data are being collected about their children, and it is unclear why it is being collected, how it is being used, and what it could be used for in the future. Rising distrust between parents, stakeholders and technology providers shuts down constructive conversation and only serves to exacerbate the issue.

4. ROLE OF THE EDM COMMUNITY

The barriers to industry applications of educational data mining techniques stem from several sources. Existing and proposed policy put restrictions on how data can be collected, stored and used. Companies can technically comply with legislation without much impact on their product or processes. However, strictly adhering to policies and offering real privacy protection often makes accessing and using educational tools more difficult, giving less socially responsible companies a competitive advantage. Public opinion can lead to the destruction of companies with no unethical practices and can drive money away from investment in data-based educational technologies. The EDM community has an important role to play in keeping these challenges in check and allowing innovation to thrive (Table 1).

4.1 Transparency

A lack of clarity, rampant misunderstanding, and a high degree of uncertainty fuel sentiment against the collection and use of student data. The main concerns of many parents and privacy advocates are largely not reflective of actual practice.

Consequently, the EDM community is unique positioned to advance public understanding for what student data is really being used. EDM professionals can better describe how data is being used, what innovations it supports, explain the focus of current research, and portray likely research foci of the field. Parental concerns may be allayed knowing that people are not actively contributing to the outcomes they most fear.

The community can also disseminate details about the effectiveness of these approaches beyond the research community. Showing the strengths of these techniques may help concerned individuals see the benefits that individual children and the education system as a whole stand to gain.

As new approaches are developed, consider creating public-facing talking points that can be used to communicate with concerned parties. These points should describe what data is being used and

Table 1. The role of the EDM community on the issue of student privacy.

Point of Concern	Proposed Solution	Action Item
Policy	• Policy Activism	• Remain abreast of proposed or approved policy changes. • Actively voice expert opinions to policy makers.
Corporate Social Responsibility	• Awareness of classroom constraints	• Develop algorithms that minimize the amount of data needed to produce effective results where possible. • Avoid requirements for individual accounts when possible.
Public Opinion	• Understanding public opinion • Transparency	• Actively work to correct misconceptions about student data and privacy concerns. • Set research agendas aimed at better understanding public understanding of privacy issues.

how it can benefit students. They should be written in a way that is clear and easy for non-experts to understand.

4.2 Research Agendas

The EDM community can also drive research towards areas that may help compliance with legislation and improve public opinion. Algorithms that minimize the amount of data needed to produce effective results would be beneficial to companies wishing to keep privacy concerns at bay. Researchers should consider the tradeoffs when developing new "big data" approaches. More data may lead to more effective techniques but it also may represent an increased violation of privacy. Finding a balance can support widespread dissemination in commercial technologies

It is important that researchers understand the classroom constraints of commercial educational technologies, especially when it comes to privacy. For a variety of reasons it is often less feasible to guarantee that data comes from a specific individual. Approaches that are robust enough to take this into account will allow educational technologies to be successful in more environments.

An additional area of research that could benefit from the involvement of the EDM community is research on the public understanding of privacy issues. The EDM community could be involved in cross-disciplinary research to ensure that communication surrounding EDM techniques is accurate and clear, and organizational privacy policies are widely understood.

4.3 Policy Activism

Finally, we encourage members of the EDM community to become active as policy debates grow. It is important to stay up to date on proposed policy changes and to consider how these changes may impact research agendas and the commercial applicability of those findings. Policy changes may increase constraints in commercial applications that could drive shifts in funding made available to EDM research. The policy changes affect both communities.

The discussion also needs more contributions from EDM experts. Consider voicing concerns to local officials and provide guidance on how policy should be directed. Too much of the current dialogue is based on a fear and misunderstanding. These voices are currently overpowering the experts who support the use of data in education.

5. CONCLUSION

Educational data mining offers significant promise in improving student learning and education systems as a whole. However, these systems are often driven by the collection of large amounts of student data, which is a growing concern to many. Shifts in public opinion and policy have led to barriers to the adoption of EDM technologies in commercial applications and threaten to stifle future innovation. Several fundamental issues are driving this trend.

The first is the role of trust, fear, and misunderstanding. It is difficult to combat the fear associated with the unknown. Companies and experts in the field must work hard to both gain the trust of the public and communicate what is actually being done with student data. Trust must extend the other way as well. Companies need to trust that by being open about their practices they will not be attacked by concerned external stakeholders. Fear from companies about the reactions of privacy advocates encourages silence on their parts and serves to reduce overall transparency. Both parties must build trust to move towards an open and productive dialogue.

Another recurring theme centers on legislation that has not yet had the desired effect. Privacy advocates view current legislation as too lenient and many companies are able to comply without actually protecting student data. In fact, the legislation may actually harm companies that do the most to protect student privacy. Voluntary pledges offer one solution, though they are not without problems; conflicts of interest often erode even the best self-policing strategies. Many, if not most, companies may support the spirit of such pledges but be unable to sign due to any number various technicalities. Active involvement from all invested parties will be crucial to designing new legislation that will strike a balance between allowing data to be used for the good of education, while protecting the privacy of individual students.

Finally, differing views on the appropriateness of private institutions delivering public goods underscore many of the issues discussed. If commercial vendors are going to be the major providers of educational technologies to school systems there needs to be a shift in how the public perceives these companies. Stifling the success of these companies only serves to keep innovative learning technologies out of the classroom. Still, deference to privacy concerns is an important component of occupying a role in part characterized by public stewardship. Discussions about the ethical limits of financially profiting off of student data need to be addressed directly by corporate, research, and public interests with adequate emphasis on risk and potential system improvements.

Overall, there are variety of issues contributing to concerns over student privacy and how these concerns impact industry applications of educational data mining. These issues are extremely prominent and are not expected to lose momentum soon. The EDM community stands to play an important role in how discussions and legislation around student privacy evolve in the coming years. The landscape of educational data and privacy will continue to shift, and we hope with increased involvement this shift will be positive for researchers and industries interested in using educational data mining to support student learning.

6. REFERENCES

[1] G. Siemens and R. S. J. Baker, "Learning Analytics and Educational Data Mining: Towards Communication and Collaboration," pp. 252–254, 2012.

[2] S. Simon, "Data Mining Your Children," *Politico*, 15-May-2014.

[3] N. Singer, "With Tech Taking Over in Schools, Worries Rise," *The New York Times*, 14-Sep-2014.

[4] S. Trainor, "Student data privacy is cloudy today, clearer tomorrow," *Phi Delta Kappan*, vol. 96, no. 5, pp. 13–18, 2015.

[5] B. Herold, "inBloom to Shut Down Amid Growing Data-Privacy Concerns," *Education Week*, 04-Feb-2014.

[6] "FACT SHEET: Safeguarding American Consumers & Families," *The White House*, 2015. [Online]. Available: http://www.whitehouse.gov/the-press-office/2015/01/12/fact-sheet-safeguarding-american-consumers-families.

[7] J. McQuiggan and A. W. Sapp, *Implement, Improve and Expand Your Statewide Longitudinal Data System: Creating a Culture of Data in Education*. 2014.

[8] Mayer-Schonberger and K. Cukier, *Big Data*. New York, New York: Haughton Mifflin Harcourt Publishing Company, 2013.

[9] U. S. D. of Education, "Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief," 2012.

[10] J. Manyika, M. Chui, D. Farrel, S. Van Kuiken, P. Groves, and E. Almasi, "Open data: Unlocking Innovation and Performance with Liquid Information," 2013.

[11] V. Aleven and K. Koedinger, "Limitations of Student Control: Do Students Know When They Need Help?," in *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, 2000, pp. 292–303.

[12] C. Conati, N. Jaques, and M. Muir, "Understanding attention to adaptive hints in educational games: an eye-tracking study," *Int. J. Artif. Intell. Educ.*, vol. 23, pp. 136–161, 2013.

[13] M. Wen, D. Yang, and C. Rose, "Sentiment Analysis in MOOC Discussion Forums: What does it tell us?," in *Proceedings of the 7th International Conference on Educational Data Mining*, 2014, pp. 257–260.

[14] R. S. J. Baker, A. T. Corbett, and V. Aleven, "More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing," *Knowl. Creat. Diffus. Util.*, pp. 406–415, 2008.

[15] V. Shute, "Stealth Assessment in Computer-Based Games to Support Learning," in *Computer Games and Instruction*, 2011, pp. 503–523.

[16] J. P. Rowe, L. R. Shores, B. W. Mott, and J. C. Lester, "Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments," *Int. J. Artifical Intell. Educ.*, vol. 21, no. 1–2, pp. 115–133, 2011.

[17] J. Sabourin, L. R. Shores, B. W. Mott, and J. C. Lester, "Predicting Student Self-Regulation Strategies in Game-Based Learning Environments," in *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, 2012.

[18] H. Chen, "Identifying Early Indicators for College Readiness," 2007.

[19] L. Pappano, "Using Research to Predict Great Teachers," *Harvard Education Letter*, 2011.

[20] M. Sieber, J. Tolich. "Planning ethically responsible research" Sage Publications, 2012.

[21] S. Carey, "Students, Parents and the School Record Prison A Legal Strategy for Preventing Abuse.pdf," *J. Law Educ.*, vol. 3, p. 365, 1974.

[22] T. L. Elliott, D. Fatemi, and S. Wasan, "Student Privacy Rights — History , Owasso , and FERPA," *J. High. Educ. Theory Pract.*, vol. 14, no. 4, 2014.

[23] J. R. Reidenberg, N. C. Russell, J. Kovnot, T. B. Norton, and R. Cloutier, "Privacy and Cloud Computing in Public Schools," 2013.

[24] R. Silverblatt, "Hiding behind ivory towers: Penalizing schools that improperly invoke student privacy," *Georgetown Law J.*, vol. 101, pp. 493–517, 2013.

[25] B. Smith and J. Mader, "Protecting Students' Privacy - By Law," *Sci. Teach.*, vol. 81, no. December, 2014.

[26] Children's Online Privacy Protection Act of 1998, 5 U.S.C. 6501-6505.

[27] A. Allen, "Minor Distractions: Children, Privacy and E-commerce," Houston Law Review, 2001.

[28] J. Mayfield, "Revised Children's Online Privacy Protection Rule Goes Into Effect Today Federal Trade Commission," *Federal Trade Commission*, 01-Jul-2013.

[29] Data Quality Campaign, "2014 Student Data Privacy Bills," 2014.

[30] E. Brown, "Obama to propose new student privacy legislation," *The Washington Post*, Washington D.C., 19-Jan-2015.

[31] S. Simon, "Barack Obama to seek limits on student data mining," *Politico*, 11-Jan-2015.

[32] H. Tsukayama, "More than 70 companies just signed a pledge to protect student data priacy - with some notable exceptions," *The Washington Post*, 12-Jan-2015.

[33] D. Banisar, "Privacy and data protection around the wolrd," in 21st International Conference on Privacy and Personal Data Protection, 1999.

[34] G. Yee, "Security and Privacy in Distance Education," in Information Security and Ethics: Concepts, Methodologies, Tools, and Applications, 1st ed., H. Namati, Ed. 2007, p. 4110.

[35] D. Boyd, "Response to COPPA Rule Review, 16 CFR part 312, Project No. P-104503," Washington D.C., 2011.

[36] N. S. B. Association, "Data in the Cloud: A Legal and Policy Guide for School Boards on Student Data Privacy in the Cloud Computing Era," Alexandria, VA, 2014.

[37] C. S. Media, "Student Privacy Survey," 2014.

[38] S. Fox, "In the Matter of COPPA Rule Review, 16 CFR Part 312, Project No. P-104503," Washington D.C., 2011.

[39] J. Palfrey, D. Boyd, and U. Gasser, "How the COPPA, as Implemented, Is Misinterpreted by the Public: A Research Perspective," 2010.

[40] Pew Research Center, "What Internet Users Know about Technology and the Web," 2014.

[41] C. DeLorme, "Response to COPPA Rule: Comments to be placed on the public record," Washington D.C., 2012.

[42] M. Madden, S. Cortesi, U. Gasser, A. Lenhart, and M. Duggan, "Parents, Teens, and Online Privacy," 2012.

[43] B. L. Wilcox, D. Kunkel, J. Cantor, P. Dowrick, S. Linn, and E. Palmer, "Report of the APA Task Force on Advertising and Children," 2004.

[44] K. Vanlehn, "The Behavior of Tutoring Systems," *Int. J. Artif. Intell. Educ.*, vol. 16, no. 3, pp. 227–265, 2006.

Beyond Prediction: First Steps Toward Automatic Intervention in MOOC Student Stopout

Jacob Whitehill
Harvard University
jacob_whitehill@harvard.edu

Joseph Williams
Harvard University
joseph_jay_williams@harvard.edu

Glenn Lopez
Harvard University
glenn_lopez@harvard.edu

Cody Coleman
MIT
colemanc@mit.edu

Justin Reich
Harvard University
justin_reich@harvard.edu

ABSTRACT

High attrition rates in massive open online courses (MOOCs) have motivated growing interest in the automatic detection of student "stopout". Stopout classifiers can be used to orchestrate an intervention before students quit, and to survey students dynamically about why they ceased participation. In this paper we expand on existing stop-out detection research by (1) exploring important elements of classifier design such as generalizability to new courses; (2) developing a novel framework inspired by control theory for how to *use* a classifier's outputs to make intelligent decisions; and (3) presenting results from a "dynamic survey intervention" conducted on 2 HarvardX MOOCs, containing over 40000 students, in early 2015. Our results suggest that surveying students based on an automatic stopout classifier achieves higher response rates compared to traditional post-course surveys, and may boost students' propensity to "come back" into the course.

1. INTRODUCTION

Massive open online courses (MOOCs) enable students around the world to learn from high-quality educational content at low cost. One of the most prominent characteristics of MOOCs is that, partly due to the low cost of enrollment, many students may casually enroll in a course, browse a few videos or discussion forums, and then cease participation [12, 6, 10]. Some MOOCs offer the ability to receive a "certificate" by completing a minimum number of assignments or earning enough points, and for the most part the number of students who certify in MOOCs is far lower than the number of students who register. This is not necessarily a problem – students may enroll for different reasons, not everyone cares about formal certification, and if students learn anything from a MOOC, that is arguably an important gain.

On the other hand, the fact that most students who enroll in a MOOC do not complete the course still warrants further investigation. For example, there may be some students who genuinely intended to complete a course when they enrolled but, upon encountering the lecture materials, quiz problems, or even other students, felt discouraged, frustrated, or bored, and then stopped participating in the course. Indeed, Reich [11] found that, of students who completed HarvardX pre-course surveys and expressed the *intent* to complete the course, only 22% of such students actually did so. A deeper understanding of the reasons why students stop out of a course could help course developers improve course content.

HarvardX, Harvard's strategic initiative for online education, is interested in understanding students' learning experiences in order to improve both online and residential education. Some of the questions we are currently tackling include *who* is enrolling in HarvardX courses, *why* are they enrolling, and *how* can we improve their educational experiences. In particular, we would like to know whether students stop out of HarvardX courses for reasons exogenous to their course experience – e.g., increased stress at work – or whether they quit because they disliked something about the course, especially things that course developers might be able to improve. One step towards answering this question, which we instituted starting in 2014, was to request of every student who enrolled in a HarvardX course to answer a *post-course survey*, which asks whether they liked the course and how it could be improved. Unfortunately, this effort was largely unsuccessful: response rates to these surveys were very low (around 2% of all course registrants, and less than 1% of students who had stopped out) and heavily biased toward students who had already persisted through weeks of voluntary challenges and were likely very satisfied with the course. It seems that the traditional approach to course evaluation – asking all students to evaluate a course at its end – is unlikely to work in a MOOC context.

One possible reason for the low response rate from students who stopped out is that such students quickly disengage after leaving the course, so that the likelihood of responding to a survey weeks or even months after they quit is small. Indeed, we found (see Fig. 1) that the probability of responding to (i.e., starting, but not necessarily completing) the post-course surveys decays rapidly as the time since stopout increases. It is possible that higher response rates could be achieved if students could be contacted, through some automatic mechanism, in a more timely fashion. This could potentially increase the amount of information that

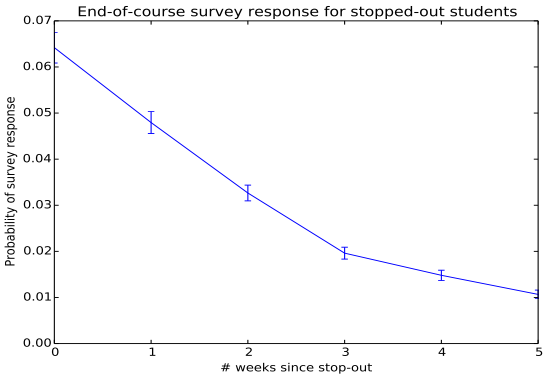


Figure 1: Mean probability (± std.err.) of responding to the post-course survey versus time-since-stopout, over 6 HarvardX MOOCs.

HarvardX, and other MOOC providers, can glean from students who choose not to complete their courses.

In January-April 2015, we pursued this idea of a *dynamic survey mechanism* designed specifically to target students who recently "stopped out". In particular, we developed an automatic *classifier* of whether a student s has "stopped out" of a course by time t. Our **definition of stopout** derives from the kinds of students we wish to survey: we say a student s has *stopped out* by time t if and only if: s does not subsequently earn a certificate *and* s takes no further action between time t and the course-end date when certificates are issued. The rationale is that students who *either* certify in a course *or* continue to participate in course activities (watch videos, post to discussion forums, etc.) can reasonably be assumed to be satisfied with the course; it is the *rest* of the students whom we would like to query. In addition to developing a stopout classifier, we developed a survey *controller* that decides, based on the classifier's output, whether or not to query student s at time t; the goal here is to maximize the rate of survey response while maintaining a low spam rate, i.e., the fraction of students who had not stopped out but were incorrectly classified as having done so (false alarms). In our paper we describe our approaches to developing the classifier and controller, as well as our first experiences in querying students and analyzing their feedback. To a modest extent, even just emailing students with "Returning to course?" in the subject line (see Sec. 6) constitutes a small "intervention"; the architecture we develop for deciding which students to contact may be useful for researchers developing automatic mechanisms for preventing student stopout.

Contributions: (1) Most prior work on stopout detection focuses on training detectors for a *single* MOOC, without examining generalization to *new* courses. For our purpose of conducting dynamic surveys and interventions, generalization to new MOOCs is critical. We thus focus our machine learning efforts on developing features that predict stopout over a wide variety of MOOCs and conduct analyses to measure cross-MOOC generalization accuracy. (2) While a variety of methods have been investigated for *detecting* stopout, almost no prior research has explored how to *use* a stopout detector to survey students or conduct an intervention. We present a principled method, based on optimization via simulation, to choose a threshold on the classifier's output so as to maximize a performance criterion. Finally, (3) we conduct one of the first MOOC "survey interventions" using an automatic stopout classifier (to our knowledge, the only other work is [7]) and report initial findings.

2. STRUCTURE OF HARVARDX MOOCS

Most HarvardX MOOCs (all those which are analyzed in this paper) are hosted on servers owned and managed by edX, which is a non-profit multi-university consortium located in Cambridge, Massachusetts. Student enrollment and event data are stored at edX and then transferred periodically (daily and weekly depending on the dataset) from edX to HarvardX. Hence, there is a "time gap" between when students generate events and when these event data are available at HarvardX.

Every HarvardX MOOC has a *start date*, i.e., the first day when participation in the MOOC (e.g., viewing a lecture, posting to the discussion forum) is possible. HarvardX MOOCs also have an *end date* when certificates are issued. At the end date, all students whose grade exceeds a minimum *certification threshold* G (which may differ for each course) receive a certificate. HarvardX courses allow students to register even after the course-end date, and they may view lectures and read the discussion forums; in most MOOCs these students cannot, however, earn a certificate. For the analyses in this paper we normalize the start date for each course to be 0 and denote the end date as T_e.

3. RELATED WORK

Over the past 3 years, since MOOCs have proliferated and the low proportion of students who complete them has become apparent, researchers from a variety of fields, including computer science, education, and economics, have begun developing quantitative models of when and why student stop out from MOOCs. The motivation for such work varies – some researchers are more interested in estimating the relative weight of different causes of stopout, whereas others (including ourselves) are primarily interested in developing automatic classifiers that could be used for real-time interventions. Work on stopout/dropout detection in MOOCs varies along several dimensions, described below:

Definition of stopout/dropout: Some researchers treat a student's last "event" within a MOOC as the stopout/dropout date, where "event" could be submitting an assignment or quiz solution [14, 13], watching a video [13], posting to a discussion forum [17], or any event whatsoever [8, 1]. Others define stopout as not earning a certificate within a course [5, 2, 4]. Hybrid definitions, such as having watched fewer than 50% of the course's videos and having executed no action during the last month [3], are also possible. Our own "stopout" definition (see Introduction) is a hybrid of lack of certification and last event.

Features used for prediction: The most commonly used features are derived from *clickstream data* [4, 1, 8, 2, 3, 14, 7] (e.g., when students play videos, post to discussion forums, submit answers to quiz problems), *grades* [4, 5, 3, 14, 7]

(e.g., average grade on quizzes), and *social network analysis* [17, 5] (e.g., eigenvector centrality of a node in a discussion forum graph). Biographical information (e.g., job, age) has also been used [5, 13, 17].

Classification method: Most existing work uses standard supervised learning methods such as support vector machines [8] and logistic regression [4, 5, 14, 7]; the latter has the advantage of probabilistic semantics and readily interpretable feature coefficients. Another approach is to use a generative model such as a Hidden Markov Model [1]; this could be useful for control-theoretic approaches to *preventing* stopout. Survival analysis techniques such as the Cox proportional hazards model have also been used [17, 13].

Classification setting: A critical issue is whether a stopout detector is highly tuned to an existing course that will never be offered again; whether it could generalize to a future offering of the same course; or whether it could generalize to other courses. Detectors that are tuned to perform optimally for only a single course are useful for exploring different classification architectures and features, but their utility for predicting stopout in new students is limited (since typically the entire course has ended before training even begins). Most existing work focuses on a single MOOC (which may or may not be offered again); to our knowledge, only [7, 3] explore stopout detection across multiple courses.

To our knowledge, the only prior work that explores how to use a stopout detector to conduct dynamic surveys is [7]. In contrast to their work, we take a more formal optimization approach to deciding how to use the classifier's output to make intelligent survey decisions (see Sec. 5).

4. STOPOUT DETECTOR

The first step toward developing our dynamic survey system is to train a classifier of student stopout. In particular, we wish to estimate the probability that a student s has stopped out by time t, given the event history up to time t. We focus on *time invariant* classifiers, i.e., classifiers whose input/output relationship is the same for all t. (An alternative approach, which we discuss in Sec. 4.3, is to train a separate classifier for each week, as was done in [14].) In correspondance with the interventions that we conduct (see Sec. 6), we vary t over $\mathcal{T} = \{10, 17, 24, \ldots, T_\mathrm{e}\}$ days; these days correspond to the timing of the survey interventions that we conduct. In our classification paradigm, if a student s stops out at time $t = 16$, then the label for s at $t = 10$ would be negative (since he/she had not yet stopped out), and the labels for times $17, 24, \ldots, T_\mathrm{e}$ would all be positive. Note that, since students may enroll at different times during the course (between 0 and T_e), not all values of t are represented for all students.

For classification we use multinomial logistic regression (MLR) with an L_2 ridge term (10^{-4}) on every feature except the "bias" term (which has no regularization). Prior to classifier training, features are normalized to have mean 0 and variance 1; the same normalization parameters (mean, standard deviation) are also applied to the testing set. For each course, we assign each student to either the training (50%) or testing (50%) group based on a hash of his/her username; hence, students who belong to the testing set for one course will belong to the testing set for *all* courses. For all experiments, we include all students who enrolled in the MOOC prior to the course-end date when certificates are issued.

As accuracy metric we use Area Under the Receiver Operating Characteristics Curve (AUC) statistic, which measures the probability that a classifier can discriminate correctly between two data points – one positive, and one negative – in a two-alternative forced-choice task [15]. An AUC of 1 indicates perfect discrimination whereas 0.5 corresponds to a classifier that guesses randomly. The AUC is *threshold independent* because it averages over all possible thresholds of the classifier's output. For a *control* task in which we use the classifier to make decisions, we face an additional hurdle of how to select the threshold (see Sec. 5).

4.1 Features

Our focus is on finding features that are predictive of stopout for a wide variety of MOOCs, rather than creating specialized features (via intensive feature engineering [14]) that are tailored to a particular course. We extract these features from two tables generated by edX: the "tracking_log" table (containing event data), and the "courseware student module" table (containing grades). The features we extract and the motivation for them are listed below:

1. The absolute time (in days, since course start) t, as well as the relative time through the course (t/T_e) – it is possible that students who persist through most of the course are unlikely to stop out.

2. The elapsed time between the last recorded event and time t – recent activity is likely negatively correlated with stopping out.

3. The total number of events of different types that were triggered by the student up to time t, where event types includes forum posts, video plays, etc.

4. 1-D temporally-local band-pass (Gabor [9]) filters (6 frequencies, 3 bandwidths) of all event times before t. Temporal Gabor filters capture sinusoidal patterns (with frequency $F = 2^f, f \in \{-10, -9, \ldots, -5\}$ days) in the *recent* history of events by attenuating with a Gaussian envelope (with bandwidth $\sigma \in \{14, 28, 56\}$ days); see Fig. 5 for examples. Gabor filters have been used previously for automatic event detection (e.g., [16]), and it is possible that "regularity" in event logs is predictive of whether a student stops out.

5. The student's grade at time t relative to the certification threshold (g_t/G), as well as a binary feature encoding whether the student already has enough points to certify ($\mathbb{I}[g_t \geq G]$). If the latter feature equals 1, then by definition the student has not stopped out.

See Appendix for more details. Including a "bias" feature (constant 1), this amounts to 37 features.

4.2 Experiments

We investigated the following questions:

Course ID	Year	Subject	# students	# certifiers	# events	# data	# + data
AT1x	2014	Anatomy	971	60	384747	7588	5895
CB22x	2013	Greek Heroes	34615	1407	11017890	671894	555581
CB22.1x	2013	Greek Heroes	17465	731	5195716	250205	201836
ER22x	2013	Justice	71513	5430	16256478	1209515	926067
GSE2x	2014	Education	37382	3936	13474171	209097	159639
HDS1544.1x	2013	Religion	22638	1546	6837110	144233	108848
PH525x	2014	Public Health	18812	652	5567125	124592	96836
SW12x	2013	Chinese History	18016	3068	7638660	78821	50431
SW12.2x	2014	Chinese History	9265	2137	3544666	25885	15741
USW30x	2014	History	14357	1089	2171359	107789	86043

Table 1: MOOCs for which we trained stopout classifiers, along with # students who enrolled up till the course-end date, # students who earned a certificate, # events generated by students up till the course-end date, # data points (summed over all students and all times t when classification was performed) for training and testing, and # positively labeled data points (time-points after the student had stopped out).

1. **Accuracy within-course**: How much variation in accuracy is there from course to course? How does this accuracy vary over $t \in [0, T_e]$ within each course?
2. **Accuracy between-courses**: How well does a classifier trained on the largest course in Table 1 (ER22x) perform on the other courses?
3. **Training set size & over/under-fitting**: Does accuracy improve if more data are collected? Is there evidence of over/under-fitting?
4. **Feature selection**: Which features are most predictive of stopout? How much accuracy is gained by adding more features?
5. **Confidence**: Does the classifier become more confident as the time-since-stopout increases?

4.3 Accuracy within-course

For this experiment we trained a separate classifier for each of 10 HarvardX MOOCs (see Table 1) using only training data and then evaluated on testing data. Accuracy for each course as a function of time-to-course-end ($T_e - t$) is shown in Fig. 2. In this graph we observe substantially lower accuracy during the beginning of each course (left side of the graph) than at the end, suggesting that longer event histories (larger t) yield more accurate classifications. In addition, accuracy varies considerably from course to course, especially at the beginning of each course.

Table 2 (middle column) shows accuracy for each course aggregated over all $t \in \mathcal{T}$. Comparing classification architectures across different courses is approximate at best; however, we do observe a large performance gap between our numbers and the accuracy reported in [1] (AUC=0.71), who also use "last event" as their definition of stopout. One possible explanation is the lack of a "time since last event" feature (see Sec. 4.6) in their feature set. [8] use a similar definition of stopout but only report percent-correct, not AUC.

Based on Fig. 2, it is conceivable that students' behavior (or the set of students) is qualitatively different during the first week of a course compared to later weeks, and that training a specialized classifier to predict stopout only during the first week might perform better than a classifier trained on all weeks' data. We explored this hypothesis in a follow-up study (ER22x only) and found minor evidence to support it: train on week 1, test on week 1 gives an AUC of 0.69; train on all weeks, test on week 1 gives an AUC of 0.66.

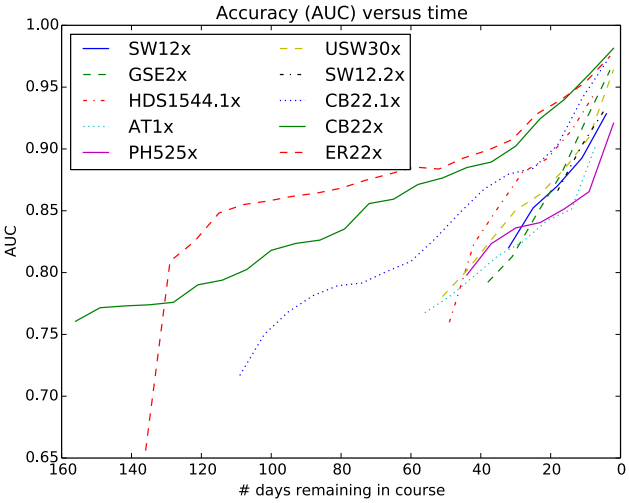


Figure 2: Accuracy (area under the receiver operating characteristics curve (AUC)) of the various stopout classifiers as a function of time, expressed as number of days until the course-end date.

4.4 Accuracy between-courses

Here, we consider only the classifier for course ER22x, containing the largest number of students and the most training data. We assessed how well the ER22x stopout classifier generalized to other courses compared to training a custom classifier for each course. We assess accuracy over all students and all $t \in \mathcal{T}$ to obtain an overall AUC score for each course. Results are shown in Table 2. The middle column shows testing accuracy when training on each course, whereas the right column shows testing accuracy when trained on ER22x. Interestingly, though a small consistent performance gain can be eked by training a classifier for each MOOC, the gap is quite small, typically < 0.02. This suggests that the features described in Sec. 4.1 are quite

Course	Within-course	Cross-train (ER22x)
AT1x	0.850	0.832
CB22x	0.879	0.876
CB22.1x	0.868	0.866
ER22x	0.895	0.895
GSE2x	0.892	0.881
HDS1544.1x	0.897	0.887
PH525x	0.860	0.847
SW12x	0.890	0.880
SW12.2x	0.907	0.896
USW30x	0.884	0.875

Table 2: Accuracy (AUC, measured over all students in the test set and all times t) of stopout classification for each course, along with accuracy when cross-training from course ER22x.

general; on the other hand, it also points to the possibility of underfitting (see Sec. 4.5).

4.5 Training set size & over/under-fitting

We examined how testing accuracy (AUC) increases as the number of training data increases. For ER22x, we found that, even if the number of training students is drastically reduced to 1000 (down from around 36000), the testing accuracy is virtually identical at 0.894. Moreover, the *training* accuracy for a training set of 1000 students is only 0.91 (and slightly lower when using the full training set) and does not improve by reducing the ridge term. These numbers suggest that (a) the feature space may be too impoverished (underfitted) to classify all data correctly; and/or (b) there is a large amount of inherent uncertainty in a student's future action given only his/her event logs and grades.

4.6 Feature selection

While some insight into feature salience can be gleaned by examining the regression coefficients, in practice it is difficult to interpret these coefficients because the L_2 regularizer distributes weight across multiple correlated features. We thus used the following greedy feature selection procedure: Initialize a feature set $\mathcal{F}$ to contain only the "bias" feature; find the feature (not already in $\mathcal{F}$) that maximally increases the AUC on training data (for ER22x); add this feature to $\mathcal{F}$ and record the associated AUC score; repeat $N-1$ times.

We executed this procedure for $N = 5$ rounds and obtained the results in Table 3. The most predictive feature was time-since-last-action (which corroborates a similar result in [7]); using this feature alone (along with the "bias" feature), the AUC was already 0.867. The student's normalized grade (g_t/G) was the second most predictive feature; this is intuitive since our definition of stopout includes certification as one of the criteria. Next, time into the course (t) was selected, suggesting there are certain times in the course when students are more likely to stop out. The fourth feature selected was a Gabor feature; rather than capturing periodicity in a student's events, the high bandwidth ($\sigma = 56$ days) and low frequency ($F = 2^{-10}$ days) of the feature can more aptly be described as a weighted sum of event counts favoring the recent past more than the distant past (see Fig. 5).

Top 5 Most Predictive Features

#	Feature	Cumulative AUC (training)
1	Time since last event	0.867
2	Normalized grade (g_t/G)	0.880
3	Time into course (t)	0.886
4	Gabor ($\sigma = 56, F = 2^{-10}$ days)	0.889
5	Total # events	0.890

Table 3: The top 5 most predictive features and associated cumulative AUC on *training* data, for ER22x. Feature i is chosen so as to maximize the training AUC given the previously selected features $1, \ldots, i-1$.

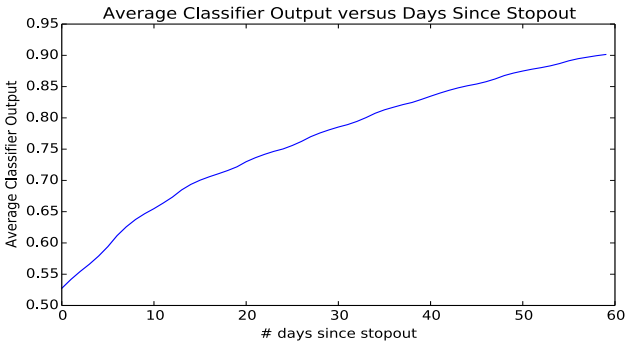


Figure 3: The average output of the ER22x stopout classifier, as a function of time-since-stopout, on students who had stopped out of the course.

In retrospect, it is clear that "time since last event" would be salient – the longer it has been since a student has done anything, the less likely he/she is to do anything in the future. It may be useful, in future stopout detection research, to compare with this single feature as a baseline.

4.7 Confidence

When building a real-time system that uses the probability estimates given by a classifier to make decisions, it can be useful to "wait" before acting until the classifier becomes more confident (so as to avoid false alarms). For course ER22x, we found that the expected classifier output at time t, averaged over every student who stopped out at time $t' < t$, increases with time-since-stopout $(t - t')$. The Pearson correlation of the classifier output y with $t - t'$ was 0.73, and the Spearman rank correlation was even higher (0.93). A graph displaying the expected classifier output versus time-since-stopout is shown in Fig. 3.

5. CONTROLLER

Given a trained classifier of student stopout, how can we use it to decide which students to contact and when to contact them? At each week t, the classifier estimates for each student s the probability y_{st} that the student has stopped out. How high must y_{st} be in order to justify querying that student at that time? In this decision problem, we are faced with the following **trade-off**:

Factor 1: The sooner we contact a student after he/she has stopped out, the higher the probability that he/she will respond (see Fig. 1); this suggests using a lower threshold.
Factor 2: On the other hand, the longer we wait after he/she has stopped out, the more accurate our classifier becomes (see Fig. 3); this suggests using a higher threshold.

Depending on how the "response fall-off curve" (factor 1) and the "confidence increase" curve (factor 2) are shaped, it is possible that a more efficient (higher response rate, lower spam rate) system can be constructed if the threshold θ on the classifier's output is chosen carefully. Factor 2 was estimated in Sec. 4.7. Factor 1 can be roughly estimated using response rate data collected from the *post-course* surveys (see Introduction) and back-dating when students who responded to the survey had stopped out.

In collaboration with the HarvardX course creation teams, we also decided on additional constraints: (1) each student can be contacted during the course at most once (so as to avoid irking students with multiple email messages), and (2) the fraction of students whom we query but who had not actually stopped out (false alarms) should not exceed $\alpha = 20\%$. Note that this false alarm rate, which is computed over students' entire trajectories through the MOOC, is different from the false alarm rate of *classification* described in Sec. 4, which is computed at multiple timepoints within each trajectory. Subject to these constraints, we wish to choose a threshold θ (a scalar) on the classifier's output y_{st} so as to *maximize* the rate of survey response from students who had stopped out. Our approach to tackling this problem is based on *optimization via simulation*.

Optimization via simulation: We built a simulator of how students generate events, what grades they earn, and when they stop out, based on historical data from prior HarvardX MOOCs. We can also simulate whether a student who stopped out at time t' responds to a survey given at time t using the "response fall-off curve" described above. Then, for any given value of θ, we can estimate how many query responses and how many false alarms it generates by averaging over many runs (we chose $N = 50000$) of the simulator: for each run, we randomly choose a student s from our training set, and at each time point t (every 7 days until T_e), we extract a feature vector x_t based on s's event log and grade up to time t. We then classify x_t using a trained classifier (from Sec. 4) and threshold the result y_{st} using θ. If $y_{st} > \theta$ *and* if we had not previously queried s during the current simulation run, then we query the student. If the student had indeed stopped out before t, then we sample the student's response (reply, not reply) from the response fall-off curve. During all simulation runs we maintain counts of both false alarms and hits (stopped-out student replies to query). Since θ is a scalar, we can use simple grid-search to find θ^* that maximizes the hit rate subject to a false alarm rate below α. Note that more sophisticated controllers with multidimensional parameter vectors θ are also possible (e.g., a different threshold for every week of the MOOC) using policy gradient optimization methods.

6. SURVEY INTERVENTION

Using the classifier and controller described above, we conducted a "dynamic survey intervention" on two live HarvardX courses: HLS2x ("ContractsX") and PH525x ("Statistics and R for the Life Sciences"), which started on Jan. 8 and Jan. 19, 2015, respectively. The goals were to (1) collect feedback about why stopped-out students left the course and (2) explore how sending a simple survey solicitation email affects students' behavior.

We trained separate stopout classifiers, using previous HarvardX courses for which stopout data were already available, for HLS2x and PH525x. For PH525x, there was a 2014 version of the course on which we could train. For HLS2x, we trained on a 2014 course ("AT1x") whose lecture structure (e.g., the frequency with which lecture videos were posted) was similar. Then, using each trained classifier and the response fall-off curve estimated from post-course survey data (see Sec. 5), we optimized the classifier threshold θ for each MOOC ($\theta = 0.79$ for HLS2x, $\theta = 0.75$ for PH525x).

We emailed students in batches once per week. Each week, we ran the stopout classifier on all students who had registered and were active in the course (i.e., had not de-registered). Each student was assigned a condition (50% experimental, 50% control) based on a hash of his/her username. To every student s in the experimental group whose y_{st} at time t exceeded θ, we sent an email (see Fig. 4) asking whether he/she intended to complete the course and why/why not. After clicking on a link, the user is given the opportunity to enter free-response feedback in a textbox. We used Qualtrics to manage the surveys, send the emails, and track the results. Students in the control group were not emailed; instead, we used them to measure the accuracy of our stopout classifier and to compare the "comeback rates" across conditions.

We delivered 3 batches (Jan. 21, Jan. 26, Feb. 2) of survey emails to 5073 students in HLS2x and 1 batch (Feb. 2) to 3764 students in PH525x. These dates were chosen to occur shortly after the data transfers from edX to HarvardX (see Sec. 2). Except in Sec. 6.2, we exclude students (138 (2.7%) from HLS2x, 201 (5.4%) from PH525x) from our analyses whom we *would not have emailed* if we had had real-time access to students' event data. Hence, the results below estimate the response rates, accuracy, and comeback rates if we could run our intervention directly on edX's servers (with 0 time-gap).

6.1 Response rate from stopped-out students

We investigated whether the dynamic survey intervention induced more stopped-out students to respond compared to the conventional post-course survey mechanism. Because the HarvardX post-course surveys are much longer than our stop-out survey, we compared the rates with which stopped-out students *started* the surveys (without necessarily completing them) to enable a fairer comparison. We analyzed response rates for HLS2x only (PH525x is still ongoing).

To measure response rates, we computed the number of students D whom we emailed *and* who had actually stopped out (which we now know since the course has ended) before the email was sent. Then, of these D students, we compute the number N of students who responded to (started, but not necessarily completed) the survey, and then calculated the response rate N/D. Since the last intervention for HLS2x was on Feb. 2, which was 32 days before the course-end date

```
Dear Jake,

We hope you have enjoyed the opportunity to explore ContractsX. It has been a while since you logged into the course,
so we are eager to learn about your experience. Would you please take this short survey, so we can improve the course for
future students? Each of the links below connects to a short survey. Please click on the link that best describes you.
[ ]  I plan on continuing with the course
[ ]  I am not continuing the course because it was not what I expected when I signed up.
[ ]  I am not continuing the course because the course takes too much time.
[ ]  I am not continuing the course because I am not happy with the quality of the course.
[ ]  I am not continuing the course because I have learned all that I wanted to learn.
[ ]  I am not continuing the course now, but I may at a future time.
Your feedback is very important to us. Thank you for registering for ContractsX.
```

Figure 4: A sample email delivered as part of our dynamic survey intervention for HLS2x.

(Mar. 6), we also calculated the corresponding fraction of students in previous HarvardX courses who responded to the post-course surveys who had stopped out at least 32 days before the course-end date (c.f. Fig. 1).

Result: The response rate from stopped-out students for the dynamic survey intervention was 3.7% compared to 1.0% for the post-course survey mechanism; the difference was statistically significant ($\chi^2(1) = 183, p < 10^{-15}$, 2-tailed). In other words, the dynamic survey mechanism achieved over 3x higher response rate.

6.2 Survey responses

For this analysis we included *all* students whom we emailed (even those whom we would not have emailed with real-time data; see above). From HLS2x, 336 students (6.6%) responded to (i.e., started but not necessarily finished) the survey. From PH525x, 353 students (9.4%) responded to the survey. Note that, in contrast to [7], who reported a 12.5% response rate for a computer science MOOC, we did not condition on students having watched at least one video.

Of students who started the survey *and* answered whether or not they planned to continue (329 for HLS2x, 328 for PH525x), most replied that they planned to continue the course (242 for HLS2x, 203 for PH525x). Of those who replied they did *not* wish to continue (87 for HLS2x, 125 for PH525x), the reasons are broken down as follows:

Reason	Freq.
"It was not what I expected when I signed up"	8.4%
"The course takes too much time"	5.0%
"I am not happy with the quality of the course"	0.5%
"I have learned all that I wanted to learn"	5.5%
"I may at a future time"	80.7%

In other words, many respondents who confirmed they had stopped out indicated that they also might resume the course in the future. Notably, very few respondents reported that the courses were of poor quality. However, we emphasize that the full population of registrants who stop out could potentially be very different from the sample who responded to the survey; hence, the numbers above should be interpreted with caution. Our stopout detector may disproportionately identify students who stop out because they are too busy, or students who stop out because they are too busy may disproportionately respond to our survey and students unhappy with the course may choose not to respond.

6.3 Accuracy

As a further assessment of the stopout detector described in Sec. 4, we computed the accuracy of the classifier on students in the control group of our HLS2x intervention.

Results: The accuracy (AUC) for HLS2x was 0.74 for week 1, 0.78 for week 2, and 0.80 for week 3. These numbers are consistent with the results in Sec. 4.3.

6.4 Effect on student "comeback"

One survey respondent wrote: "I was not allocating time for edX, but receiving your survey e-mail recaptured my attention." This raises the question of whether the mere act of notifying students that we believed they had lost interest might cause them to "come back". To test this hypothesis, we compared the fraction of students in the experimental group who "came back" – i.e., took at least one action (other than de-registering and/or responding to the survey) in the course after we sent the emails – to the corresponding fraction of students in the control group. We assessed comeback rates at two different timepoints – Feb. 12 (before we submitted the paper for review) and Apr. 20 (before we submitted the paper for final publication) – using all event data available by those dates.

Results: For all 4 interventions (3 weeks of HLS2x, and 1 week of PH525x), the comeback rates were higher at both timepoints for the experimental group (who received an email) than for the control group (who did not receive an email). Aggregated over all weeks of both courses, the comeback rate by Feb. 12 was 12.4% for the experimental group versus 11.2% for the control group; the difference was statistically significant ($\chi^2(1) = 5.63, p = 0.018$, 2-tailed). By Apr. 20, however, the difference was smaller – 22.1% for the experimental group versus 21.4% for the control group – and not statistically significant ($\chi^2(1) = 1.25, p = 0.26$, 2-tailed).

Together, these results suggest that the intervention induced students to come back *sooner* into the course, even if the overall comeback rates are similar. To confirm this hypothesis, we compared the mean "comeback time" (time between last action before intervention, and first action after intervention, among students who came back) between the two groups and across all 4 interventions. We found that students in the experimental group came back significantly sooner: 51.68 days for the experimental group versus 55.02 days for the control group (Mann-Whitney $U = 1458393, n_1 = 1725, n_2 = 1831, p < 10^{-4}$, 2-tailed). These

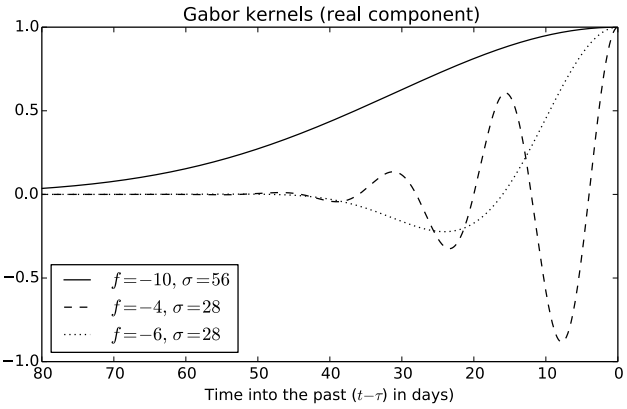


Figure 5: Sample Gabor kernels.

results provide evidence that an "intervention" consisting of an email indicating that a student has been flagged as having potentially stopped out, can affect students' behavior.

7. CONCLUSIONS

We developed an automatic classifier of MOOC student "stopout" and showed that it generalizes to new MOOCs with high accuracy. We also presented a novel end-to-end architecture for conducting a "dynamic survey intervention" on MOOC students who recently stopped out to ask them why they quit. Compared to post-course surveys, the dynamic survey mechanism attained a significantly higher response rate. Moreover, the mere act of asking students why they had left the course induced students to "come back" into the course more quickly. Preliminary analysis of the surveys suggest students quit due to exogenous factors (not enough time) rather than poor quality of the MOOCs.

Limitations: The subset of stopped-out students who responded to the survey may not be a representative sample; thus, results in Sec. 6.2 should be interpreted with caution.

Future work: In future work we will explore whether more sophisticated, time-*variant* classifiers such as recurrent neural networks can yield better performance. With more accurate classifiers we can conduct more efficient surveys and more effective interventions to reduce stopout.

APPENDIX

Event count features: We counted events of the following types (using the "event_type" field in the edX "tracking_log" table): "showanswer", "seek_video", "play_video", "pause_video", "stop_video", "show_transcript", "page_close", "problem_save", "problem_check", and "problem_show". We also measured activity in discussion forums by counting events whose "event_type" field contained "threads" or "forum".

Gabor features: A Gabor filter kernel (see Fig. 5) is the product of a Gaussian envelope and a complex sinusoid. At time $t-\tau$ (i.e., τ days before t), the real and imaginary components are given by $K_{\mathrm{r}}(\tau) = \exp(-\pi\tau^2/(2\sigma^2))\cos(2\pi F\tau)$ and $K_{\mathrm{i}}(\tau) = \exp(-\pi\tau^2/(2\sigma^2))\sin(2\pi F\tau)$ (respectively), where σ is the bandwidth of the Gaussian envelope and F is the frequency of the sinusoid. When extracting Gabor features at time t, we convolve this complex kernel with a t-dimensional "history vector" h whose τth component contains the total number of events generated by that student on day $t-\tau$. We then compute the magnitude of the complex filter response, i.e., $\left|\sum_{\tau=1}^{t}\left(K_{\mathrm{r}}(\tau)h_\tau + jK_{\mathrm{i}}(\tau)h_\tau\right)\right|$, where $j=\sqrt{-1}$.

8. REFERENCES

[1] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. Technical report, UC Berkeley, 2013.

[2] C. Coleman, D. Seaton, and I. Chuang. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Learning at Scale*, 2015.

[3] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in MOOCs using learner activity features. In *European MOOC Summit*, 2014.

[4] J. He, J. Bailey, Benjamin, I. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *AAAI*, 2015.

[5] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O'Dowd. Predicting MOOC performance with week 1 behavior. In *Educational Data Mining*, 2014.

[6] H. Khalil and M. Ebner. MOOCs completion rates and possible methods to improve retention - a literature review. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2014.

[7] R. Kizilcec and S. Halawa. Attrition and achievement gaps in online learning. In *Learning at Scale*, 2015.

[8] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting MOOC dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, 2014.

[9] J. Movellan. Tutorial on Gabor filters. Technical report, UCSD Machine Perception Laboratory, 2002.

[10] D. Onah, J. Sinclair, and R. Boyatt. Dropout rates of massive open online courses: behavioural patterns. In *Conf. on Education and New Learning Tech.* , 2014.

[11] J. Reich. MOOC completion and retention in the context of student intent. *EDUCAUSE Review*, 2014.

[12] R. Rivard. Measuring the MOOC dropout rate. *Insider Higher Ed*, 2013.

[13] R. Stein and G. Allione. Mass attrition: An analysis of drop out from a principles of microeconomics MOOC. *PIER Working Paper*, 14(031), 2014.

[14] C. Taylor, K. Veeramachaneni, and U.-M. O'Reilly. Likely to stop? Predicting stopout in massive open online courses. *arXiv*, 2014. `http://arxiv.org/abs/1408.3382`.

[15] C. Tyler and C.-C. Chen. Signal detection theory in the 2AFC paradigm: attention, channel uncertainty and probability summation. *Vision Research*, 40(22):3121–3144.

[16] E. Vural, M. Bartlett, G. Littlewort, M. Cetin, A. Ercil, and J. Movellan. Discrimination of moderate and acute drowsiness based on spontaneous facial expressions. In *ICPR*, 2010.

[17] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. "Turn on, tune in, drop out": Anticipating student dropouts in massive open online courses. In *NIPS Workshop on Data-Driven Education*, 2014.

From Predictive Models to Instructional Policies

Joseph Rollinson
Computer Science Department
Carnegie Mellon University
jtrollinson@gmail.com

Emma Brunskill
Computer Science Department
Carnegie Mellon University
ebrun@cs.cmu.edu

ABSTRACT

At their core, Intelligent Tutoring Systems consist of a student model and a policy. The student model captures the state of the student and the policy uses the student model to individualize instruction. Policies require different properties from the student model. For example, a mastery threshold policy requires the student model to have a way to quantify whether the student has mastered a skill. A large amount of work has been done on building student models that can predict student performance on the next question. In this paper, we leverage this prior work with a new when-to-stop policy that is compatible with any such predictive student model. Our results suggest that, when employed as part of our new predictive similarity policy, student models with similar predictive accuracies can suggest that substantially different amounts of practice are necessary. This suggests that predictive accuracy may not be a sufficient metric by itself when choosing which student model to use in intelligent tutoring systems.

1. INTRODUCTION

Intelligent tutoring systems offer the promise of highly effective, personalized, scalable education. Within the ITS research community, there has been substantial work on constructing student models that can accurately predict student performance (e.g. [6, 3, 15, 5, 10, 9, 14, 7]). Another key issue is how to improve student performance through the use of instructional policy design. There has been significant interest in cognitive models used for within activity design (often referred to as the inner-loop) and even authoring tools developed to make designing effective activities easier (e.g. CTAT [1]). However, there has been much less attention to outer-loop (what problem to select or when to stop) instructional policies (though exceptions include [5, 12, 17]).

In this paper we focus on a common outer-loop ITS challenge, adaptively deciding when to stop teaching a certain skill to a student given correct/incorrect responses. Somewhat surprisingly, there are no standard policy rules or algorithms for deciding when to stop teaching for many of the student models introduced over the last decade. Bayesian Knowledge Tracing [6] naturally lends itself to mastery teaching, since one can halt when the student has mastered a skill with probability above a certain threshold. Such a mastery threshold has been used as part of widely used tutoring systems, but typically in conjunction with additional rules since a student may never reach a sufficient mastery threshold given the available activities.

We seek to be able to directly use a wide range of student models to create instructional policies that halt both when a student has learned a skill and when the student seems unlikely to make any further progress given the available tutoring activities. To do so we introduce an instructional policy rule based on change in predicted student performance.

Our specific contributions are as follows:

- We provide a functional interface to student models that captures their predictive powers without knowledge of their internal mechanics (Section 3).
- We introduce the *predictive similarity policy*: a new when-to-stop policy that can take as input any predictive student-model (Section 4) and can halt both if students have successfully acquired a skill or do not seem able to do so given the available activities.
- We analyze the performance of this policy compared to a mastery threshold policy on the KDD dataset and find our policy tends to suggest similar or a smaller number of problems than a mastery threshold policy (Section 5).
- We also show that our new policy can be used to analyze a range of student models with similar predictive performance (on the KDD dataset) and find that they can sometimes suggest very different numbers of instructional problems. (Section 5).

Our results suggest that predictive accuracy alone can mask some of the substantial differences among student models. Polices based on models with similar predictive accuracy can make widely different decisions. One direction for future work is to measure which models produce the best learning policies. This will require new experiments and datasets.

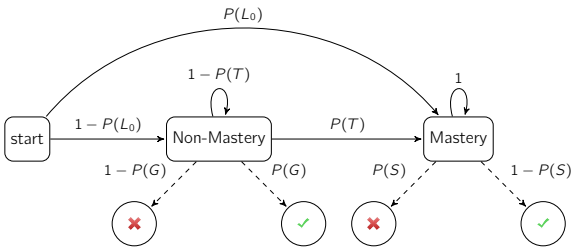


Figure 1: BKT as a Markov process. *Mastery* and *Non-Mastery* are hidden states. Arrow values represent the probability of the transition or observation.

2. BACKGROUND: STUDENT MODELS

Student models are responsible for modeling the learning process of students. The majority of student models are *predictive models* that provide probabilistic predictions of whether a student will get a subsequent item correct. In this section we describe two popular predictive student models, *Bayesian knowledge tracing* and *latent factor models*. Note that other predictive models, such as Predictive State Representations (PSRs), can also be used to calculate the probability of a correct response [7].

2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [6] tracks the state of the student's knowledge as they respond to practice questions. BKT treats students as being in one of two possible hidden states: *Mastery* and *Non-Mastery*. It is assumed that a student never forgets what they have mastered and if not yet mastered, a new question always has a fixed static probability of helping the student master the skill. These assumptions mean that BKT requires only four trained parameters:

$P(L_0)$	Initial probability of mastery.
$P(T)$	Probability of *transitioning* to mastery over a single learning opportunity.
$P(G)$	Probability of *guessing* the correct answer when the student is not in the mastered state.
$P(S)$	Probability of *slipping* (making a mistake) when the student is in the mastered state.

After every response, the probability of mastery, $P(L_t)$, is updated with Bayesian inference. The probability that a student responds correctly is

$$P_{\text{BKT}}(C_t) = (1 - P(S))P(L_t) + P(G)(1 - P(L_t)). \quad (1)$$

Prior work suggests that students can get stuck on a particular activity. Unfortunately, BKT as described above assumes that students will inevitably master a skill if given enough questions. As this is not always the case, in industry BKT is often used together with additional rules to make instructional decisions.

2.2 Latent Factor Models

Unlike BKT models, Latent Factor Models (LFM) do not directly model learning as a process [3]. Instead, LFMs assume that there are latent parameters of both the student and skill that can be used to predict student performance. These parameters are learned from a dataset of students answering questions on multiple skills. The probability that the student responds correctly to the next question is calculated by applying the sigmoid function to the linear combination of parameters p and features f.

$$P_{\text{LFM}}(C) = \frac{1}{1 - e^{-f \cdot p}} \quad (2)$$

Additive Factor Models (AFM) [3] are based on the assumption that student performance increases with more questions. A student is represented by an aptitude parameter (α_i) and a skill is represented by a difficulty parameter (β_k) and learning rate (γ_k). AFM is sensitive to the number of questions the student has seen, but ignores the correctness of student responses. The probability that student i will respond correctly after n responses on skill k is

$$P_{\text{AFM}}(C) = \frac{1}{1 - e^{-(\alpha_i + \beta_k + \gamma_k n)}}. \quad (3)$$

Performance Factor Models (PFM) [15] are an extension of AFMs that are sensitive to the correctness of student responses. PFMs separate the skill learning rate into success and failure parameters, μ_k and ρ_k respectively. The probability that student i will respond correctly after s correct responses and f incorrect responses on skill k is

$$P_{\text{PFM}}(C) = \frac{1}{1 - e^{-(\alpha_i + \beta_k + \mu_k s + \rho_k f)}}. \quad (4)$$

LFMs can easily be extended to capture other features. For example, the instructional factors model [5] extends PFMs with a parameter for the number of tells (interactions that do not generate observations) given to the student. To our knowledge there is almost no work on using LFMs to capture temporal information about the order of observations. Unlike BKT, LFMs are not frequently used in instructional policies.

Though structurally different, BKT models, AFMs and PFMs tend to have similar predictive accuracy [9, 15]. This raises the interesting question of whether instructional policies that use these models are similar.

3. WHEN-TO-STOP POLICIES

We assume a simple intelligent tutoring system that teaches students one skill at a time. All questions are treated the same, so the system only has to decide when to stop providing the student questions. In this section, we provide a general framework for the when-to-stop problem. In particular, we describe an interface that abstracts out the student model from instructional policies, which we will use to define the MASTERY THRESHOLD policy and use in the next section as the foundations of a model-agnostic instructional policy.

3.1 Accessing Models

Policies require a mechanism for getting values from student models to make decisions. We describe this mechanism as a state type and a set of functions. A student model consists of two types of values: immutable parameters that are learned on training data and mutable state that changes over time. For example, the parameters for BKT are $(P(L_0), P(T), P(G), P(S))$ and the model state is the probability of mastery $(P(L_t))$. Policies treat the state

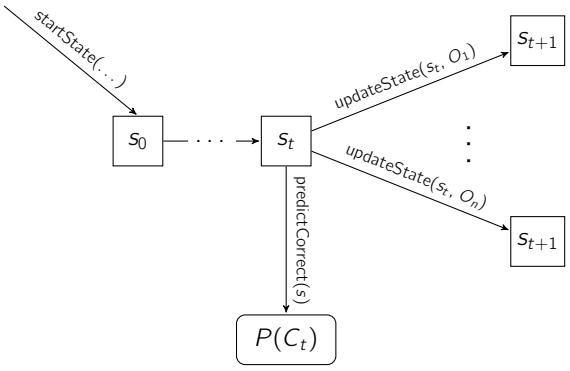


Figure 2: Model process with functional interface

as a black box, which they pass to functions. All predictive student models must provide the following functions. **startState**(...) returns the model state given that the student has not seen any questions. **updateState**(state, obs) returns an updated state given the observation. For this paper, observations are whether the student got the last question correct or incorrect. Finally, predictive student models must provide **predictCorrect**(state), which returns the probability that the student will get the next question correct. The function interfaces for BKT models and PFM are provided in table 1. Under this abstraction, when-to-stop policies are functions **stop**(state) that output true if the system should stop providing questions for the current skill and false if the system should continue providing the student with questions.

3.2 Mastery Threshold Policy

The MASTERY THRESHOLD POLICY halts when the student model is confident that the student has mastered the skill. This implies that we want to halt when the student masters the skill. Note that if the estimate of student mastery is based solely on a BKT[1] then a mastery threshold policy implicitly assumes that every student will master the skill given enough problems. Mathematically, we want to stop at time t if $P(L_t) > \Delta$, where Δ is our mastery threshold. The MASTERY THRESHOLD policy function can be written as:

$$\textbf{stop}_M(\text{state}) = \textbf{predictMastery}(\text{state}) > \Delta. \quad (5)$$

The MASTERY THRESHOLD policy can only be used with models that include **predictMastery**(state) in their function set. BKT models are compatible, but LFMs are not. By itself, the MASTERY THRESHOLD does not stop if the student has no chance of attaining mastery in the skill with the given activities. Students on poorly designed skills could be stuck learning a skill indefinitely.

4. FROM PREDICTION TO POLICY

In educational data mining, a large emphasis is put on building models that can accurately predict student observations. Our goal is to build a new when-to-stop policy that will work with any predictive student model.

[1]In practice, industry systems that use mastery thresholds and BKTs often use additional rules as well.

Our new instructional policy is based on a set of assumptions. First, students working on a skill will eventually end in one of two hidden end-states. Either, they will master the skill, or they will be unable to master the skill given the activities available. Second, once students enter either end-state, the probability that they respond correctly to a question stays the same. Third, if the probability that a student will respond correctly is not changing, then the student is in an end-state. Finally, we should stop if the student is in an end-state.

From these assumptions it follows that if the probability that the student will respond correctly to the next question is not changing, then we should stop. In other words, we should stop if it is highly likely that showing the student another question will not change the probability that the student will get the next question correct by a significant amount. We propose to stop if

$$(P(|P(C_t) - P(C_{t+1})| < \epsilon)) > \delta \quad (6)$$

where $P(C_t)$ is the probability that the student will get the next question right. This can be thought of as a threshold on the sum of the probabilities of each observation that will lead to an insignificant change in the probability that a student will get the next question correct, which can be written as

$$\sum_{o \in O} P(O_t = o)\mathbb{1}(|P(C_t) - P(C_{t+1}|O_t = o)| < \epsilon) > \delta \quad (7)$$

where $P(C_{t+1}|O_t = o)$ is the probability that the student will respond correctly after observation o, O_t is the observation at time t, and $\mathbb{1}$ is an indicator variable. In our case $O = \{C, \neg C\}$. This expression is true in the following cases:

1. $P(C_t) > \delta$ and $|P(C_t) - P(C_{t+1}|C_t)| < \epsilon$
2. $P(\neg C_t) > \delta$ and $|P(C_t) - P(C_{t+1}|\neg C_t)| < \epsilon$
3. $|P(C_t) - P(C_{t+1}|C_t)| < \epsilon$ and $|P(C_t) - P(C_{t+1}|\neg C_t)| < \epsilon$

First, if a student is highly likely to respond correctly to the next question and the change in prediction is small if the student responds correctly, then we should stop. Second, if a student is highly unlikely to respond correctly to the next question and the change in prediction is small if the student responds incorrectly, then we should stop. Third, if the change in prediction is small no matter how the student responds, then we should stop. All terms in these expressions can be calculated from the predictive student model interface as shown in equations 8 and 9. We call the instructional policy that stops according to these three cases the PREDICTIVE SIMILARITY policy. The function for the PREDICTIVE SIMILARITY policy is provided in algorithm 1

$$P(C_t) = \textbf{predictCorrect}(s) \quad (8)$$

$$P(C_{t+1}|O_t) = \textbf{predictCorrect}(\textbf{updateState}(s, O_t)) \quad (9)$$

5. EXPERIMENTS & RESULTS

We now compare the PREDICTIVE SIMILARITY policy to the MASTERY THRESHOLD policy and see if using different student models as input to the predictive SIMILARITY POLICY yields quantitatively different policies.

Table 1: Functional interfaces for BKT and PFM

	BKT	PFM
startState$(\dots)$	$P(L_0)$	$(\alpha_i + \beta_k, \mu_k, \rho_k, 0, 0)$
updateState(s, o)	$P(L_{t+1}\|P(L_t), O_{t+1} = o)$	$\begin{cases}(w, \mu, \rho, s+1, f) & \text{if } o = C\\ (w, \mu, \rho, s, f+1) & \text{if } o = \neg C\end{cases}$
predictCorrect(s)	$P(\neg S)P(L_t) + P(G)(1 - P(L_t))$	$\left(1 + e^{-(w+s\mu+f\rho)}\right)^{-1}$
predictMastery(s)	$P(L_t)$	—

Algorithm 1 PREDICTIVE SIMILARITY policy stop function

```
1: function STOP(state)
2:     P(C_t) ← predictCorrect(state)
3:     total ← 0
4:     if P(C_t) > 0 then
5:         state' ← updateState(state, correct)
6:         P(C_{t+1}|C_t) ← predictCorrect(state')
7:         if |P(C_t) − P(C_{t+1}|C_t)| < ε then
8:             total ← total + P(C_t)
9:     if P(C_t) < 1 then
10:        state' ← updateState(state, incorrect)
11:        P(C_{t+1}|¬C_t) ← predictCorrect(state')
12:        if |P(C_t) − P(C_{t+1}|¬C_t)| < ε then
13:            total ← total + (1 − P(C_t))
14:    return total > δ
```

5.1 ExpOps

In order to better understand the differences between two instructional policies we will measure the expected number of problems to be given to students by a policy using the ExpOps algorithm. The ExpOps algorithm allows us to summarize an instructional policy into a single number by approximately calculating the expected number of questions an instructional policy would provide to a student. A naive algorithm takes in the state of the student model and returns 0 if the instructional policy stops at the current state or recursively calls itself with an updated state given each possible observation as shown in equation 10. It builds a synthetic tree of possible observations and their probability using the model state. The tree grows until the policy decides to stop teaching the student. This approach does not require any student data nor does it generate any observation sequences. However, this algorithm may never stop, so ExpOps approximates it by also stopping if we reach a maximum length or if the probability of the sequence of observations thus far drops below a path threshold as shown in algorithm 2. In this paper, we use a path threshold of 10^{-7} and a maximum length of 100.

$$E[Ops] = \begin{cases} 0 & \text{if } \textbf{stop}(s) \\ 1 + \sum_{o \in O} P(O_t = o) E[Ops|o] & \text{otherwise} \end{cases} \quad (10)$$

Lee and Brunskill first introduced this metric to show that individualized models lead to significantly different policies than the general models [12].

5.2 Data

Algorithm 2 Expected Number of Learning Opportunites

```
1: function EXPOPS(startState)
2:     function EXPOPS'(state, P(path), len)
3:         if P(path) < pathThreshold then
4:             return 0
5:         if len ≥ maxLen then
6:             return 0
7:         if stop(state) then
8:             return 0
9:         P(C) ← predictCorrect(state)
10:        P(W) ← 1 − P(C)
11:        expOpsSoFar ← 0
12:        if P(C) > 0 then
13:            P(path + c) ← P(path) * P(C)
14:            state' ← updateState(state, C)
15:            ops ← EXPOPS'(state', P(path + c), len + 1)
16:            expOpsSoFar ← expOpsSoFar + (ops * P(C))
17:        if P(W) > 0 then
18:            P(path + w) ← P(path) * P(W)
19:            state' ← updateState(state, incorrect)
20:            ops ← EXPOPS'(state', P(path + w), len + 1)
21:            expOpsSoFar ← expOpsSoFar + (ops * P(W))
22:        return 1 + expOpsSoFar
23:    return EXPOPS'((startState, 1, 0))
```

For our experiments we used the Algebra I 2008–2009 dataset from the KDD Cup 2010 Educational Data Mining Challenge [18]. This dataset was collected from students learning algebra I using Carnegie Learning Inc.'s intelligent tutoring systems. The dataset consists of 8,918,054 rows where each row corresponds to a single step inside a problem. These steps are tagged according to three different knowledge component models. For this paper, we used the SubSkills knowledge component model. We removed all rows with missing data. We combined the rows into observation sequences per student and per skill. Steps attached to multiple skills were added to the observation sequences of all attached skills. We removed all skills that had less than 50 observation sequences. Our final dataset included 3292 students, 505 skills, and 421,991 observation sequences.

We performed 5-fold cross-validation on the datasets to see how well AFM, PFM, and BKT models predict student performance. We randomly separated the dataset into five folds with an equal number of observation sequences per skill in each fold. We trained AFM, PFM, and BKT models on four of the five folds and then predicted student performance on

Table 2: Root Mean Squared Error on 5 Folds

Fold	BKT	PFM	AFM
0	0.353	0.364	0.368
1	0.359	0.367	0.371
2	0.358	0.368	0.371
3	0.366	0.369	0.374
4	0.353	0.365	0.368

the leftover fold. We calculated the root mean squared error found in Table 2. Our results show that the three models had similar predictive accuracy, agreeing with prior work.

5.3 Model Implementation

We implemented BKT models as hidden Markov models using a python package we developed. We used the Baum-Welch algorithm to train the models, stopping when the change in log-likelihood between iterations fell below 10^{-5}. For each skill, 10 models with random starting parameters were trained, and the one with the highest likelihood was picked. Both AFM and PFM were implemented using scikit-learn's logistic regression classifier [16]. We used L1 normalization and included a fit intercept. The tolerance was 10^{-4}. We treated an observation connected to multiple skills as multiple observations, one per skill. It is also popular to treat them as a single observation with multiple skill parameters. In the interest of reproducibility, we have published the models used as a python package.[2]

5.4 Experiment 1: Comparing policies

The MASTERY THRESHOLD policy is frequently used as a key part of deciding when to stop showing students questions. However without additional rules, it does not stop if students cannot learn the skill from the current activities. In this experiment we compare the PREDICTIVE SIMILARITY policy to the MASTERY THRESHOLD policy to see if the PREDICTIVE SIMILARITY policy acts like the MASTERY THRESHOLD policy when students learn and stops sooner when students are unable to learn with the given tutoring. We based both policies on BKT models.

We ran ExpOps on each skill for both policies. For the MASTERY THRESHOLD policy, we used the community standard threshold of $\Delta = 0.95$. For the PREDICTIVE SIMILARITY policy, we decided that the smallest meaningful change in predictions is 0.01 and that our confidence should be 0.95, so we set $\epsilon = 0.01$ and $\delta = 0.95$. We then split the skills into those where the BKT model trained on them had semantically meaningful parameters and the rest. A BKT model was said to have semantically meaningful parameters if $P(G) \leq 0.5$ and $P(S) \leq 0.5$. 218 skills had semantically meaningful parameters and 283 did not.[3].

[2]The packages are available at `http://www.jrollinson.com/research/2015/edm/from-predictive-models-to-instructional-policies.html`.

[3]We found similar results for both experiments using BKT models trained through brute force iteration on semantically meaningful values. These results can be found at `http://www.jrollinson.com/research/2015/edm/from-predictive-models-to-instructional-policies.html`

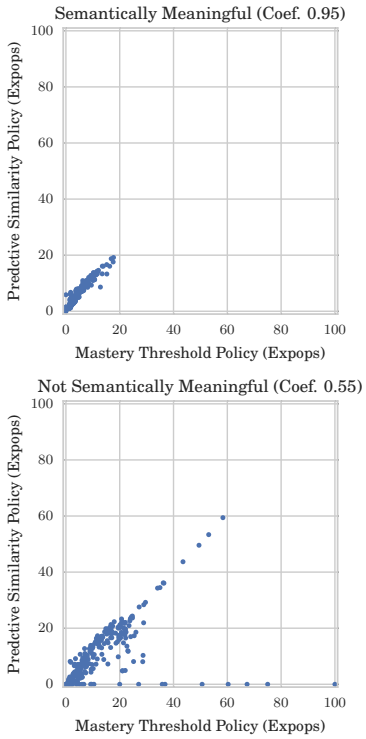


Figure 3: ExpOps using the mastery threshold policy and the predictive similarity policy on skills with and without semantically meaningful parameters.

The Pearson correlation coefficient between ExpOps values calculated using the two policies on skills with semantically meaningful parameters was 0.95. This suggests that the two policies make very similar decisions when based on BKT models with semantically meaningful parameters. However, the Pearson correlation coefficient between ExpOps values calculated using the two policies on skills that do not have semantically meaningful parameters was only 0.55. To uncover why the correlation coefficient was so much lower on skills that do not have semantically meaningful parameters, we plotted the ExpOps values calculated with the MASTERY THRESHOLD policy on the X-axis and the ExpOps values calculated with the PREDICTIVE SIMILARITY policy on the Y-axis for each skill as shown in figure 3. This plot shows that the PREDICTIVE SIMILARITY policy tends to either agree with the MASTERY THRESHOLD policy or have a lower ExpOps value on skills with parameters that are not semantically meaningful. This suggests that the PREDICTIVE SIMILARITY policy is stopping sooner on skills that students are unlikely to learn. The mastery policy does not give up on these skills, and instead teaches them for a long time.

5.5 Experiment 2: Comparing models with the predictive similarity policy

The previous experiment suggests that the PREDICTIVE SIMILARITY policy can effectively mimic the good aspects MASTERY THRESHOLD policy when based on a BKT model. We now wish to see how using models with similar predictive accuracy, but different internal structure will affect it. LFMs

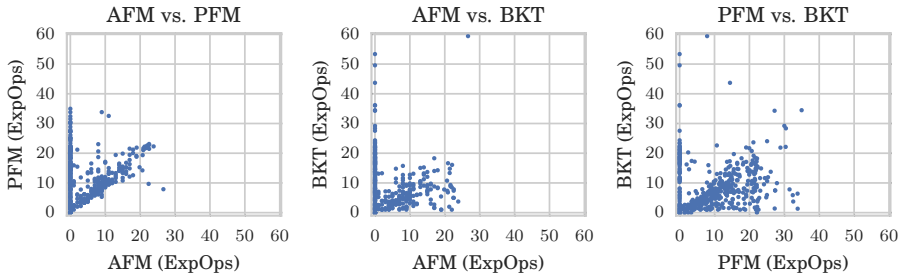


Figure 4: ExpOps plots for the predictive similarity policy using BKT, AFM, and PFM.

Table 3: Correlation coefficients on ExpOps values from policies using BKT, AFM, and PFM.

Models	Coefficient with all skills	Coefficient with skills not stopped immediately
AFM vs. PFM	0.32	0.72
AFM vs. BKT	-0.06	0.44
PFM vs. BKT	0.16	0.46

and BKT models have vastly different structure making them good models for this task. Our earlier results also found that AFM, PFM, and BKT models have similar predictive accuracy. We ran ExpOps on each skill with the PREDICTIVE SIMILARITY policy based both on AFM and PFM. AFM and PFM require a student parameter, which we set to the mean of their trained student parameters. This is commonly done when modeling a student that has not been seen before. We compared the ExpOps values for these two models with the values for the BKT-based PREDICTIVE SIMILARITY policy calculated in the previous experiment.

We first looked at how many skills the different policies immediately stopped on. We found that the BKT-based policy stopped immediately on 31 (6%) of the skills, whilst PFM stopped immediately on 130 (26%) and AFM stopped immediately on 295 (59%).

We calculated the correlation coefficient between each pair of policies on all skills as well as just on skills in which both policies did not stop immediately as shown in table 3. We found that AFM and PFM had the highest correlation coefficient. For each pair of policies, we found that removing the immediately stopped skills had a large positive impact on correlation coefficient. The BKT-based policy had a correlation coefficient of 0.44 with the AFM-based policy and 0.46 with the PFM-based policy on skills that were not immediately stopped on. This suggests that there is a weak correlation between LFM-based and BKT-based policies.

We plotted the ExpOps values for each pair of policies, shown in figure 4. The AFM vs. PFM plot reiterates that the AFM-based and PFM-based policies have similar ExpOps values on skills where AFM does not stop immediately. The BKT vs. PFM plot shows that the PFM-based policy either immediately stops or has a higher ExpOps value than the BKT-based policy on most skills.

To understand why the PFM-based policy tends to either stop immediately or go on for longer than the BKT-based policy, we studied two skills. The first skill is 'Plot point on minor tick mark — integer major fractional minor' on which the BKT-based policy has an ExpOps value of 7.0 and the PFM-based policy has an ExpOps value of 20.7. The second skill is 'Identify solution type of compound inequality using and' on which the BKT-based policy has an ExpOps value of 11.4 and the PFM-based policy immediately stops. We calculated the predictions of both models on two artificial students, one who gets every question correct and one who gets every question incorrect. In figure 5, we plot the prediction trajectories to see how the predictions of the two models compare. In both plots, the PFM-based policy asymptotes slower than the BKT-based policy. Since LFMs calculate predictions with a logistic function, PFM predictions asymptote to 0 when given only incorrect responses and 1 when given only correct responses, whereas the BKT model's predictions asymptote to $P(G)$ and $1-P(S)$ respectively. In the first plot, the PFM-based policy learns at a slower rate than the BKT-based policy, but the predictions do begin to asymptote by the 20^{th} question. In the second plot, the PFM-based policy learns much more slowly. After 25 correct questions, the PFM-based policy's prediction changes by just over 0.1, and after 25 incorrect questions, the PFM-based policy's predictions changes by less than 0.03. In contrast, the BKT-based policy asymptotes over 10 questions to $1 - P(S) = 0.79$ when given correct responses and $P(G) = 0.47$ when given incorrect responses.

This figure also shows how the parameters of a BKT model affect decision making. $P(L_0)$ is responsible for the initial probability of a correct response. $P(S)$ and $P(G)$ respectively provide the upper and lower asymptotes for the probability of a correct response. $P(T)$ is responsible for the speed of reaching the asymptotes. For the PREDICTIVE SIMILARITY policy, the distance between the initial probability of a correct response and the asymptotes along with the speed of reaching the asymptotes is responsible for the number of questions suggested.

6. DISCUSSION

Our results from experiment 1 show that the PREDICTIVE SIMILARITY policy performs similarly to the MASTERY THRESHOLD policy on BKT models with semantically meaningful parameters and suggests the same or fewer problems

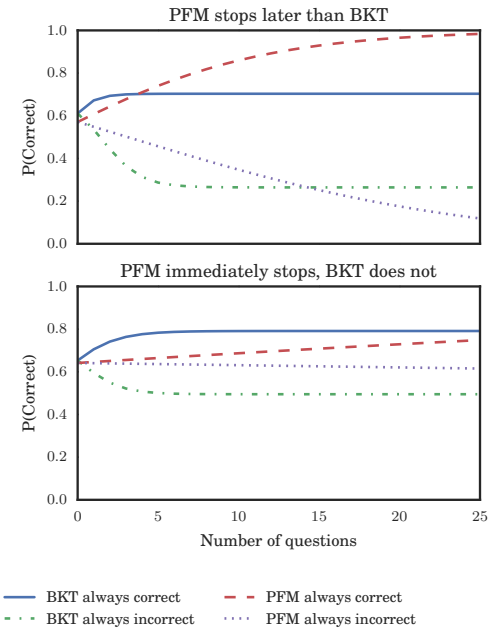


Figure 5: Predictions of BKT models and PFMs if given all correct responses or all incorrect responses on two skills.

on BKT models without semantically meaningful parameters. Thus, this experiment suggests that the two instructional policies treat students successfully learning skills similarly. The lower ExpOps values for the PREDICTIVE SIMILARITY policy provide evidence that the PREDICTIVE SIMILARITY policy does not waste as much student time as the MASTERY THRESHOLD policy on its own. Fundamentally, the MASTERY THRESHOLD policy fails to recognize that some students may not be ready to learn a skill. The PREDICTIVE SIMILARITY policy does not make the same error. Instead, the policy stops either when the system succeeds in teaching the student or when the skill is unteachable by the system. In practice MASTERY THRESHOLD policies are often used in conjunction with other rules such as a maximum amount of practice before stopping. A comparison of such hybrid policies to the PREDICTIVE SIMILARITY policy is an interesting direction for future work. However, it is important to note that such hybrid policies would still require the underlying model to have a notion of mastery, unlike our predictive similarity policy.

The PREDICTIVE SIMILARITY policy can be used to uncover differences in predictive models. Experiment 2 shows that policies based on models with the same predictive power can have widely different actions. AFMs had a very similar RMSE to both PFMs and the BKT models, but immediately stopped on a majority of the skills. An AFM must provide the same predictions to students who get many questions correct and students who get many questions incorrect. To account for this, its predictions do not change much over time. One may argue that this suggests that AFM models are poor predictive models, because their predictions hardly change with large differences in state. Both AFMs and PFMs have inaccurate asymptotes because it is likely that students who have mastered the skill will not get every question correct and that students who have not mastered the skill will not get every question incorrect. This means that these models will attempt to stay away from their asymptotes with lower learning rates. One possible solution would be to build LFMs that limit the history length. Such a model could learn asymptotes that are not 0 and 1.

7. RELATED WORK

Predictive student models are a key area of interest in the intelligent tutoring systems and educational data mining community. One recent model incorporates both BKT and LFM into a single model with better predictive accuracy than both [10]. It assumes that there are many problems associated with a single skill, and each problem has an item parameter. If we were to use such a model in a when-to-stop policy context, the simplest approach would be to find the problem with the highest learning parameter for that skill, and repeatedly apply it. However, this reduces Khajah et al.'s model to a simple BKT model, which is why we did not explicitly compare to their approach.

Less work has been done on the effects of student models on policies. Fancsali et al. [8] showed that when using the MASTERY THRESHOLD policy with BKT one can view the mastery threshold as a parameter controlling the frequency of false negatives and false positives. This work focused on simulated data from BKT models. Since BKT assumes that students eventually learn, this work did not consider wheel-spinning. Rafferty et al. [17] showed that different models of student learning of a cognitive matching task lead to significantly different partially observable Markov decision process policies. Unlike our work which focuses on deciding when-to-stop teaching a single activity type, that work focused on how to sequence different types of activities and did not use a standard education domain (unlike our use of KDD cup). Mandel et al. [13] did a large comparison of different student models in terms of their predicted influence on the best instructional policy and expected performance of that policy in the context of an educational game; however, like Rafferty et al. their focus was on considering how to sequence different types of activities, and instead of learning outcomes they focused on enhancing engagement. Chi et al. [5] performed feature selection to create models of student learning designed to be part of policies that that would enhance learning gains on a physics tutor; however, the focus again was on selecting among different types of activities rather than a when-to-stop policy. Note that neither BKT nor LFMs in their original form can be used to select among different types of problems, though extensions to both can enable such functionality. An interesting direction of future work would be to see how to extend our policy to take into account different types of activities.

Work on when-to-stop policies is also quite limited. Lee and Brunskill [12] showed that individualizing student BKT models has a significant impact on the expected number of practice opportunities (as measured through ExpOps) for a significant fraction of students. Koedinger et al. [11] showed that splitting one skill into multiple skills could significantly improve learning performance; this process was done by human experts and leveraged BKT models for the policy design. Cen et al.[4] improved the efficiency of student learn-

ing by noticing that AFM models suggested that some skills were significantly over or under practiced. They created new BKT parameters for such skills and the result was a new tutor that helped students learn significantly faster. However, the authors did not directly use AFM to induce policies, but rather used an expert based approach to transform the models back to BKT models, which could be used with existing mastery approaches. In contrast, our approach can be directly used with AFM and other such models.

Our policy assumes that learning is a gradual process. If you were to instead subscribe to an all-at-once method of learning, you could possibly use the moment of learning as your stopping point. Baker et al. provide a method of detecting the moment at which learning occurs [2]. However, this work does not attempt to build instructional policies.

8. CONCLUSION & FUTURE WORK

The main contribution of this paper is a when-to-stop policy with two attractive properties: it can be used with any predictive student model and it will provide finite practice both to students that succeed in learning a given skill and to those unable to do so given the presented activities.

This policy allowed us for the first time to compare common predictive models (LFMs and BKT models) in terms of their predicted practice required. In doing so we found that models with similar predictive error rates can lead to very different policies. This suggests that if they are to be used for instructional decision making, student models should not be judged by predictive error rates alone. One limitation of the current work is that only one dataset was used in the experiments. To confirm these results it would be useful to compare to other datasets.

One key issue raised by this work is how to evaluate instructional policy accuracy. One possible solution is to run trials with students stopping after different numbers of questions. The student would take both a pre and post-test, which could be compared to see if the student improved. However, such a trial would require many students and could be detrimental to their learning.

There is a lot of room for extending this instructional policy. First, we would like to incorporate other types of interactions, such as dictated information ("tells") or worked examples, into the PREDICTIVE SIMILARITY policy. This would give student models more information and hopefully lead to better predictions. Second, the PREDICTIVE SIMILARITY policy is myopic, and we are interested in the effects of expanding to longer horizons. Third, we are excited about extending this instructional policy to choosing between skills. Instead of stopping when there is a high probability of predictions not changing, the instructional policy could return either the skill that had the highest chance of a significant change in prediction, or the skill with the highest expected change in prediction.

9. REFERENCES

[1] V. Aleven, B. M. McLaren, J. Sewall, and K. R. Koedinger. The cognitive tutor authoring tools (ctat): preliminary evaluation of efficiency gains. In *ITS*. Springer, 2006.

[2] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting learning moment-by-moment. *IJAIED*, 21(1), 2011.

[3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *ITS*. Springer, 2006.

[4] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *FAIA*, 158, 2007.

[5] M. Chi, K. R. Koedinger, G. J. Gordon, P. W. Jordan, and K. VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *EDM*, 2011.

[6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMAP*, 4(4), 1994.

[7] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *EDM 2013*, 2010.

[8] S. E. Fancsali, T. Nixon, and S. Ritter. Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. In *EDM*, 2013.

[9] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *ITS*, 2010.

[10] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. *EDM*, 2014.

[11] K. R. Koedinger, J. C. Stamper, E. A. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *AIED*. Springer, 2013.

[12] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *EDM*, 2012.

[13] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. AAMAS, 2014.

[14] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *UMAP*. Springer, 2010.

[15] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. 2009.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12, 2011.

[17] A. Rafferty, E. Brunskill, T. Griffths, and P. Shafto. Faster teaching by POMDP planning. In *AIED*, 2011.

[18] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra 1 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge. find it at http://pslcdatashop.web.cmu.edu/kddcup/downloads.jsp.

Your model is predictive— but is it useful? Theoretical and Empirical Considerations of a New Paradigm for Adaptive Tutoring Evaluation

José P. González-Brenes
Digital Data, Analytics and Adaptive Learning
Pearson School Research
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

Yun Huang
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
yuh43@pitt.edu

ABSTRACT

Classification evaluation metrics are often used to evaluate adaptive tutoring systems— programs that teach and adapt to humans. Unfortunately, it is not clear how intuitive these metrics are for practitioners with little machine learning background. Moreover, our experiments suggest that existing convention for evaluating tutoring systems may lead to suboptimal decisions. We propose the Learner Effort-Outcomes Paradigm (Leopard), a new framework to evaluate adaptive tutoring. We introduce Teal and White, novel automatic metrics that apply Leopard and quantify the amount of effort required to achieve a learning outcome. Our experiments suggest that our metrics are a better alternative for evaluating adaptive tutoring.

Keywords

evaluation, efficacy, classification evaluation metrics

1. INTRODUCTION

A fundamental part of the scientific and engineering process is *testability*— the property of evaluating whether a hypothesis or method can be supported or falsified by data of actual experience. For example, in educational data mining, we formulate testable hypotheses that claim that the methods we engineer improve the outcomes of learners. In this manuscript, we study how to verify learner outcome hypotheses.

We focus on evaluating a popular type of educational method called *adaptive intelligent tutoring system*. Adaptive systems teach and adapt to humans; their promise is to improve education by optimizing the subset of *items* presented to students, according to their historical performance [5], and on features extracted from their activities [10]. In this context, items are questions, problems, or tasks that can be graded individually.

Evaluation metrics are important because they quantify the extent of whether an educational system helps learners. For example, a practitioner may use an evaluation method to choose which of the alternative adaptive tutoring systems to deploy in a classroom, or school district. On the other hand, a researcher may be interested in quantifying the improvements of her system compared to previous technology.

Our main contributions are proposing a novel evaluation paradigm for assessing adaptive tutoring and examples of when traditional evaluation techniques are misleading. This paper is organized as follows: § 2 reviews related methods for evaluating adaptive systems; § 3 describes the paradigm we propose for automatic evaluation of tutoring systems; § 4 provides a meta-evaluation of our novel evaluation techniques; and, § 5 provides some concluding remarks.

2. BACKGROUND

Adaptive tutoring is often implemented as a complex system with many components, such as a student model, content pool, and a cognitive model. Adaptive tutoring may be evaluated with randomized control trials. For example, in a seminal study [5] that focused on earlier adaptive tutors, a controlled trial measured the time students spent on tutoring and their performance on post-tests. The study reported that the tutoring system enabled significantly faster teaching, while students maintained the same or better performance on post-tests

Unfortunately, controlled trials can become extremely expensive and time consuming to conduct: they require institutional review board approvals, experimental design by an expert, recruiting (and often payment!) of enough participants to achieve statistical power, and data analysis. Automatic evaluation metrics improve the engineering process because they enable less expensive and faster comparisons between alternative systems. Fields that have agreed on automatic evaluation have seen an accelerated pace of technological progress. For example, the widespread adoption of the Bleu metric [15] in the machine translation community has lowered the cost of development and evaluation of translation systems. At the same time, it has enabled machine translation competitions that result in great advances of translation quality. Similarly, the Rouge metric [13] has helped the automatic summarization community transition

from expensive user studies of human judgments that may take thousands of hours to conduct, to an automatic metric that can be computed very quickly.

The adaptive tutoring community has tacitly adopted conventions for evaluating tutoring systems [6, 16, 18]. Researchers often evaluate their models with classification evaluation metrics that assess the *student model* component of the tutoring system— student models are the subsystems that forecast whether a learner will answer the next item correctly. Popular classification evaluation metrics include accuracy, log-likelihood, Area Under the Curve (AUC) of the Receiver Operating Characteristic curve, and, strangely for classifiers, the Root Mean Square Error. However, automatic evaluation metrics are intended to measure an outcome of the end user. For example, the PARADISE [22] metric used in spoken dialogue systems correlates to user satisfaction scores. Not only is there no evidence that supports that classification metrics correlate with learning outcomes; but, prior work [2] has identified serious problems with them. For example, classification metrics ignore that an adaptive system may not help learners— which could happen with a student model with a flat or decreasing learning curve [1, 20]. A decreasing learning curve implies that student performance decreases with practice; this curve is usually interpreted as a modeling problem, because it operationalizes that learners are better off with no teaching. Therefore, an adaptive tutor with a student model with a decreasing learning curve does not teach students.

Surprisingly, in spite of all of the evidence against using classification evaluation metrics, their use is still very widespread in the adaptive literature [6, 16, 18]. Moreover, there is very little research on alternative evaluation techniques. A noticeable exception is recent work on individualizing student models [12]. The authors evaluated their approach using a method called *ExpOppNeed*, which calculates the expected number of practice opportunities that learners require to master the content of the tutoring curriculum. Though their evaluation methodology is extremely interesting and promising, it was not intended to be generalizable. In the next section we extend on prior work and present a novel general paradigm for evaluating adaptive systems.

3. LEOPARD EVALUATION

Adaptive tutoring implies making a trade-off between minimizing the amount of student *effort*, by carefully personalizing the curriculum, and maximizing student *outcomes* [4]. For example, repeated practice on a skill may improve student proficiency, at the cost of a missed opportunity for teaching new material. Adequate values for student effort and outcomes respond to external expectations from the social context. For example, it is not acceptable for a tutor to minimize effort by not teaching any content at all, or to maximize outcomes by taking twenty years to teach a simple concept. The right trade off is defined by subject matter experts.

We propose the novel Learner Effort-Outcomes Paradigm (Leopard) for automatic evaluation of adaptive tutoring. At its core, Leopard quantifies the effort and outcomes of students in adaptive tutoring. Even though measuring effort and outcomes is not novel by itself, our contribution is measuring both without a randomized control trial.

- Effort: Quantifies how much practice the adaptive tutor gives to students. In this paper we focus on counting the number of items assigned to students but, alternatively, amount of time could be considered.
- Outcome: Quantifies the performance of students after adaptive tutoring. For simplicity, we operationalize performance as the percentage of items that students are able to solve after tutoring. We assume that the performance on solving items is aligned to the long-term interest of learners.

We argue that Leopard is more intuitive than classification metrics because the effort and outcome resonate to educational principles. We now describe two novel metrics that apply the Leopard philosophy. In § 3.1, we describe Teal, a metric that calculates the theoretical expected behavior of students when interacting with a family of student models; and in § 3.2, we describe White[1] a metric that uses empirical data that may have not been collected on a control trial.

3.1 Theoretical Evaluation of Adaptive Learning Systems (Teal)

We formulate Theoretical Evaluation of Adaptive Learning Systems (Teal) to evaluate adaptive tutoring from the expected behavior of their student model. Teal focuses on models of the *Knowledge Tracing Family*— a very popular set of student models [10].

To use Teal on data collected from students, we first train a model using an algorithm from the Knowledge Tracing family (§ 3.1.1), then we use the learned parameters to calculate the effort (§ 3.1.2) and outcome (§ 3.1.3) for each skill. We discuss how to use Teal on models that use features (§ 3.1.4) and our design decisions (§ 3.1.5).

3.1.1 Knowledge Tracing Family

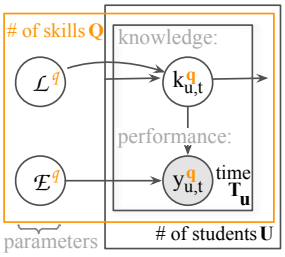


Figure 1: Knowledge Tracing plate diagram. The color of the circles represent whether the variable is latent (white), or observed in training (light), and plates represent repetition.

Figure 1 describes the Knowledge Tracing [5] model, the most simple member of the family. Knowledge Tracing requires a mapping of items to skills, often built by subject matter experts, although automatic approaches exist [8]. These skill mappings are also called cognitive models, or Q-matrices. Knowledge Tracing uses a Hidden Markov Model (HMM) per skill to model the student's knowledge as latent variables. The binary observation variable $y^q_{u,t}$ represents

[1]Tradition names metrics like colors! E.g., Rouge, Bleu.

whether the student u applies the t^{th} practice opportunity of skill q correctly. The latent variable $k^q_{u,t}$ models the latent student proficiency, which is often modeled with a binary variable to indicated mastery of the skill. To declutter notation, we may not explicitly write the indices q and u. There are two conventions for naming the skill-specific parameters of Knowledge Tracing. In the HMM tradition, the parameters are simply named transition or learning ($\mathcal{L}$), and emission ($\mathcal{E}$). In the educational tradition when using two latent states the parameters are called initial knowledge (l_0), learning (l), forgetting (f), guess (g) and slip (s). The Knowledge Tracing family includes models that parameterize the emission probabilities, transition probabilities, or both. For example, in Knowledge Tracing, the emission probability of emitting an answer $\mathbf{y}$ when the student has knowledge $\mathbf{k}$ is:

$$\mathcal{E}_{\mathbf{y},\mathbf{k}} = p(\mathbf{y}|\mathbf{k}) \tag{1}$$

Which is simply a binomial probability. To allow features in the emissions, we replace the binomial with a logistic regression [10]:

$$\mathcal{E}_{\mathbf{y},\mathbf{k}}(\boldsymbol{\beta}, \mathbf{X}_t) = p(\mathbf{y}|\mathbf{k}; \boldsymbol{\beta}, \mathbf{X}_t) \tag{2}$$

$$= \frac{1}{1 + \exp(-\boldsymbol{\beta}^\intercal \cdot \mathbf{X}_t)} \tag{3}$$

Here $\mathbf{X}_t$ is the feature vector extracted at time t, and $\boldsymbol{\beta}$ is the regression coefficient vector. The feature may indicate, for example, if the student requested a hint.

3.1.2 Effort

Teal calculates the expected number of practice that an adaptive tutor gives to students. We assume a policy that the tutor stops teaching a skill once the student is very likely to answer the next item correctly according to a model from the Knowledge Tracing Family. For notational convenience, we define the probability of answering the next item correctly as:

$$c_{t+1}(\mathbf{y}_1, \ldots, \mathbf{y}_T) \equiv p(y_{t+1} = \text{correct}|\mathbf{y}_1, \ldots, \mathbf{y}_t; \mathcal{L}, \mathcal{E}) \tag{4}$$

Here $\mathcal{L}$ and $\mathcal{E}$ are the parameters of the Knowledge Tracing Family model. We can estimate c_{t+1} using conventional inference techniques for HMMs [19], such as the Forward-Backward algorithm.

The adaptive tutor teaches an additional item if two conditions hold: (i) it is likely that the student will get the next item wrong— in other words, the probability of answering correctly the next item is below a threshold R; and (ii) the tutor has not decided to stop instruction already. More formally, the tutor keeps teaching if:

$$\text{teach}(\mathbf{y}_1, \ldots, \mathbf{y}_t, R) \equiv \begin{cases} 1 & \text{if } \underset{t'<t}{\forall} c_{t'+1}(\mathbf{y}_1, \ldots, \mathbf{y}_{t'}) < R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We now can calculate at which practice opportunity the tutor should stop instruction. For simplicity, we assume all sequences are of length T. We simply count all of the times the tutor decides to teach a new item:

$$\text{cost}_R(\mathbf{y}_1, \ldots, \mathbf{y}_T) \equiv \sum_{t=1}^{T} \text{teach}(\mathbf{y}_1, \ldots, \mathbf{y}_t, R) \tag{6}$$

Note that if the probability of answering correctly the next item has not reached the threshold in T time steps, the cost is defined as T. Teal defines effort as the expected value of the number of practice opportunities a tutor gives. This is:

$$\text{effort}(R) \equiv \mathbb{E}\left(\text{cost}_R(\mathcal{Y}_T)\right) \tag{7}$$

$$= \sum_{\mathbf{y}_1, \ldots, \mathbf{y}_T \in \mathcal{Y}_T} \underbrace{\text{cost}_R(\mathbf{y}_1, \ldots, , \mathbf{y}_T)}_{\text{amount of practice}} \cdot \underbrace{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}_{\text{sequence likelihood}} \tag{8}$$

Here, $\mathcal{Y}_T$ is the set of all sequences of length T. When we have binary student outcomes (correct or not), the cardinality of this set is 2^T, which makes Teal only tractable for sequences of a few dozens of observations. In our experience, the sequences of adaptive tutoring systems are often in this range. In a companion paper [9] we give an alternative formulation of Teal that allows approximate calculations. The likelihood of the sequence can be efficiently estimated using the Forward-Backward algorithm.

3.1.3 Outcome

We define the outcome of a student as the mean performance after the tutor should stop instruction. For a particular sequence with student cost $k = \text{cost}_R(\mathbf{y}_1, \ldots, \mathbf{y}_T)$, this is:

$$\text{outcome}(\mathbf{y}_1, \ldots, \mathbf{y}_T, k) \equiv \begin{cases} \text{mean}(y_k \ldots y_T) & \text{if } k < T \\ \text{impute value} & \text{otherwise} \end{cases} \tag{9}$$

We map the correct and incorrect student responses y_t into 1 or 1, respectively. If the student sequence does not reach the performance threshold, we impute the value of the outcome. In this paper, we set the imputation value to 0. We define the score as the expected value of the outcome:

$$\text{score}(R) \equiv \mathbb{E}(\text{outcome}(\mathcal{Y}_T, k)) \tag{10}$$

$$= \sum_{\mathbf{y}_1, \ldots, \mathbf{y}_T \in \mathcal{Y}_T} \text{outcome}(\mathbf{y}_1, \ldots, \mathbf{y}_T, R) \cdot p(\mathbf{y}_1, \ldots, \mathbf{y}_T) \tag{11}$$

3.1.4 Usage on Models With Features

For models that parameterize emission or transitions we first must build a counterfactual feature vector $\mathbf{X}$, and use it to calculate model parameters that do not depend on features. For example, consider a model that uses a binary feature vector that encodes students in different conditions. Conditions can be any feature of interest of the tutoring system, such as the ability to display multimedia content. We can use Teal to calculate the effort of students in each of the specific conditions.

For example, consider a feature vector $\mathbf{X} = (f_1, f_2, \ldots, f_n)$. Feature f_1 is 1 iff the student is using condition 1 (e.g., multimedia content is available), feature f_2 is 1, iff the student is using condition 2, etc. The vector is all zeros if the student is in the control condition. If we activate feature f_1, we can calculate the effort or score of students in the treatment 1. To apply Teal we first estimate counterfactual slip and guess parameters using Equation 3. We can use the counterfactual parameters with Teal.

For some models with features, Teal may require that students are assigned randomly to feature activation conditions, so that the regression coefficients can be interpreted as causal effects. Teal may not be appropriate if – for example – the features have reverse causality, or if there are omitted variables in the model.

3.1.5 *Design Discussion*

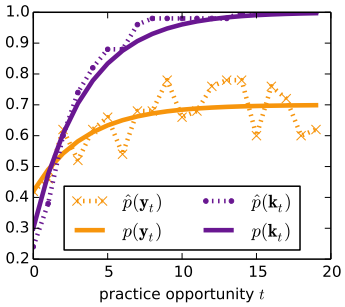


Figure 2: Expected and empirical student performance for a skill ($\ell_0 = 0.3$, $\ell = 0.25$, $g = 0.3$, $s = 0.3$, $f = 0$).

Teal extends the ExpOppNeed algorithm discussed on § 2. We compare both approaches to justify our design decisions.

1. **When to stop tutoring.** Teal expects tutoring to stop once the student is very likely to apply the skill correctly. On the other hand, ExpOppNeed relies on stopping tutoring once the posterior probability of the latent variable for knowledge is above a threshold. Figure 2 compares both approaches for some Knowledge Tracing parameters. The solid lines represent the expected values derived theoretically[2] for both strategies. To illustrate what actual student behavior may look like, we plotted dotted lines for 50 synthetic students sampled from a HMM. Although individual students vary, their average behavior is close to theoretical.

   In the figure, with 15 practice opportunities the students have close to 100% probability of skill mastery, while they only have 65% probability of applying the skill correctly. This big gap between the probability of mastery and probability of correct (the two solid lines) implies that the model is defining mastery as a state when students have low probability of applying the skill correctly. Low probability of answering correctly in a mastery state can occur due to a number of problems, for example, an incorrect item-to-skill mapping, or confusing tutoring content. We argue that an evaluation metric should penalize such models to be consistent with the Mastery Learning Theory [3].

   Moreover, prior work [1] has demonstrated that some ill-defined models have probability of correct decreasing with practice opportunities, at the same time that the probability of mastery increases. ExpOppNeed does not penalize such ill-defined models, but Teal does.

[2]Prior work derived [21]: $p(y_t = \text{correct}) = 1 - s - A\beta^t$. Here, $\beta = (1 - \ell)$, and $A = (1 - s - g) \cdot (1 - \ell_0)$

Algorithm 1 Single-Skill White

Require: performance sequences $\mathbf{y}_{u,q,t}$, student model predictions $\hat{\mathbf{c}}_{u,q,t}$ (the subscripts index students, skills, and practice opportunities), threshold R

function WHITE($\mathbf{y}_{u,q,t}, \hat{\mathbf{c}}_{u,q,t}, R$)
  for each student u **do**
    for each skill q **do**
      ▷ *Select data for student u and skill q only:*
      $\mathbf{y}', \hat{\mathbf{c}}' \leftarrow \text{filter}(\mathbf{y}, \hat{\mathbf{c}}, u, q)$
      $\text{effort}(q, u) \leftarrow 0$
      for each practice opportunity t in $\mathbf{y}'$ **do**:
        if $\hat{\mathbf{c}}'_{t+1} \geq R$ **then**
          $\text{score}(q, u) \leftarrow \text{mean}(y_{t+1}, \ldots, y_T)$
          next skill q
        else if last(t) **then**
          $\text{score}(q, u) \leftarrow$ impute
        $\text{effort}(q, u) \leftarrow \text{effort}(q, u) + 1$
  return effort, score

2. **What to measure.** ExpOppNeed does not calculate expected outcome of students. Teal considers both student outcome and effort because it is trivial to optimize one of the metrics if the other one is ignored.

3. **Precision of the results** Both ExpOppNeed and Teal have exponential computational complexity. However, ExpOppNeed uses a heuristic to prune sequences with low probability. Unfortunately, if the effort is very high (or infinite), the likelihood of the individual sequences becomes very low, and ExpOppNeed prunes the sequences too soon and therefore it may underestimate the effort. Teal improves on ExpOppNeed by defining effort on fixed-length sequences and not doing pruning.

We now summarize some limitations of our approach. Teal assumes that the model parameters are correct, and does not take into account potential modeling problems— such as misspecification, or over-fitting. By design, Teal only is able to evaluate models in the Knowledge Tracing Family. We now present a novel evaluation method that addresses these limitations.

3.2 Whole Intelligent Tutoring System Empirical Evaluation (White)

We propose Whole Intelligent Tutoring System Evaluation (White), a novel automatic method that evaluates the recommendations of an adaptive system using data. White does not assume the student data is generated by a Knowledge Tracing model; instead, it relies on counterfactual simulations. White reproduces the decisions that the tutoring system *would* have made given the input data on the test set, by counting how many items the adaptive tutor would ask students to solve, and what is the mean student performance after tutoring.

Algorithm 1 describes White for a tutoring system that assumes an item is assigned to exactly one skill. We leave more complex tutors for future work. The input of White is the student performance sequences $\mathbf{y}$, the predictions of answering correctly $\hat{\mathbf{c}}$, and a threshold R that defines what is the

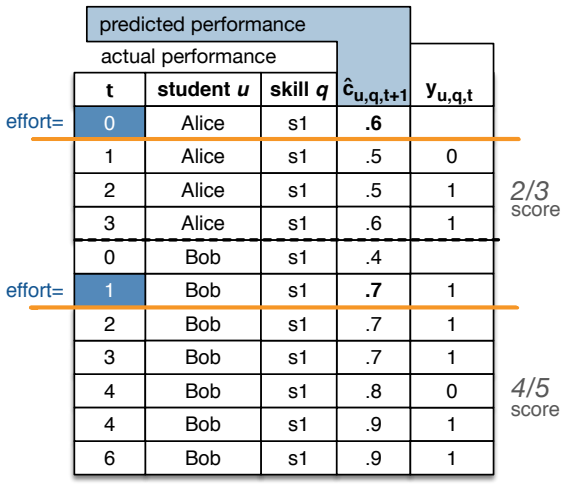


Figure 3: Example of White calculating counterfactual score and effort using empirical data ($R = 0.6$).

target probability of correct. White assumes that the students are a random sample of the student population. The predictions are calculated by the student model component of the adaptive tutoring. For a data-driven student model, the predictions can be informed with the history preceding the current time step. For instance, to predict on the third time step, the student model may use the data up to the second time step. For example, for Knowledge Tracing:

$$\hat{c}_t = \hat{p}(y_t = \text{correct}|\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}) \quad (12)$$

Figure 3 shows example data of how White works for a 60% threshold ($R = 0.6$). For each student and skill in the test set, White estimates their counterfactual effort— how many items the student *would* have solved using the tutoring system. In our example, Alice does not get to practice the skill because the student model believes that she is likely to already know it (effort =0), but Bob is given one practice opportunity (effort=1). After Bob answers correctly the item, he is not given any more practice. White also calculates a counterfactual score to represent the student learning. It is the percentage of correct answers after the instruction would have stopped. The score is related to an existing classification evaluation metric called precision. Precision aggregates the entire dataset, while score is computed by students and skills. Although superficially it may sound as a small difference, our strategy allows us to avoid a special case of the Simpson's Paradox. In § 4.1.1 we discuss the issue more.

In this paper, when we report results with White, we impute the score of students that do not reach the threshold with their average performance. This is deliberately a different imputation strategy that we use with Teal, which assigns a score of zero to students that do not reach the threshold.

4. META-EVALUATION

In this section we meta-evaluate Leopard. We experiment with data from students (§ 4.1) and simulations (§ 4.2).

We compare these sets of metrics:

- **Conventional metrics**. We use classification evaluation metrics to evaluate how the student models predict future student performance. For this, we allow student models to use the history preceding the time step we want to predict.
- **Leopard metrics**. We use the score and effort as calculated by White and Teal. For simplicity we report the average scores across skills, and the sum of the mean effort. For U students and Q skills, this is:

$$\text{dataset score}(R) = \frac{1}{Q \cdot U} \sum_{q}^{Q} \sum_{u}^{U} \text{score}(q, u) \quad (13)$$

$$\text{dataset effort}(R) = \frac{1}{U} \sum_{q}^{Q} \sum_{u}^{U} \text{effort}(q, u) \quad (14)$$

4.1 Real Student Data

We use data collected from a commercial non-adaptive tutoring system for middle school Math. Our dataset includes only the first part of the entire curriculum, and contains students from the same grade from multiple schools. It contains approximately 1.2 million observations from 25,000 students. We randomly split the dataset into three sets of students. The training and test set have 60% and 20% of the students, respectively. The remainder of the data is reserved for future experiments not described in this paper. The item bank was mapped to skills in three different ways— the *coarse* definition maps the items into 27 skills, the *fine* definition into 90 skills, and the proprietary one is not reported.

4.1.1 *Are predictive models always useful?*

Assessing an evaluation metric with real student data is difficult because we often do not know the ground truth. To get around this, we now describe a strategy to select a subset of the dataset that we know the behavior of. Our main insight is that for adaptive tutoring to be able to optimize when to stop instruction, the student performance should increase with repeated practice (the learning curve should be increasing). Our strategy consists on selecting the subset of the data where student modeling may fail, because student performance remains flat or decreases with practice.

We first train a simplified Performance Factors Analysis [17] (PFA) model. We use a logistic regression for each skill:

$$p(y^q_{u,t}) = \frac{1}{1 + \exp(\boldsymbol{\beta}^q \cdot \mathbf{X}^q))} \quad (15)$$

The dimensions of $\mathbf{X}^q$ are the count of prior correct responses of the student and an intercept. We learn the parameters of the model $\boldsymbol{\beta}^q$ using constrained optimization— the regression coefficient for the effect of prior correct responses has to be non-negative.

We only use data from the skills that have zero regression coefficient for the effect of prior correct responses (flat or decreasing learning curve). Such skills are not suitable for an adaptive tutor because the PFA student model believes that practice does not influence student performance. More concretely, this PFA model would give infinite practice to difficult skills, or no practice to easy skills. Table 1 compares the results of using White and two conventional metrics on

the test set of the selected skills. We compare with a majority class model that always predicts students answers as correct. The conventional metrics we report are the AUC, because of it's popularity, and the F-metric, because in experiments we report later correlates highly with White. For White we use a threshold of 60%. We cannot report on Teal because PFA is not part of the Knowledge Tracing Family.

Table 1: Evaluation metric comparison.

	White		conventional	
	score	effort	F	AUC
Performance Factors Analysis	.18	10.1	**.79**	**.85**
Majority Class	.18	11.2	0	.50

The AUC and F-metric results are arguably very high, indicating that the PFA model is highly predictive— yet by construction, we know that the model is *not useful* for adaptivity. The high prediction power of PFA is explained only by the intercepts of the model. That is, the predictions are based on the skill difficulty, independently of the student performance. We argue that White communicates better the unfavorable nature of the model because it reports a very low score, and only a small improvement of effort when compared to a baseline.

The problem with metrics that aggregate over the entire dataset, like the AUC and the F-metric, can be explained by Simpson's paradox— a trend that appears in different groups of data that disappears or reverses when the groups are combined. Because adaptive tutors learn a model from each skill independently, it is effectively a group of models. White and Teal evaluate each skill independently and are not susceptible to this problem. Consider the alternatives:

- Reporting as a baseline the *difficulty classifier*— a classifier that only considers the fraction of correct answers of each skill in the training set. For example, in Table 1, the PFA model has an AUC of 0.8, the same as the difficulty classifier. Because PFA did not outperform this baseline, it suggests the student model has a problem. However, simulations [8] provide evidence that useful student models may have predictive performance similar to the difficulty classifier. Therefore, the difficulty classifier baseline may reject some useful student models. Moreover, convention expects classifiers to have an AUC of higher than 0.5 to be useful, and this new baseline would break this interpretation.
- Calculating classification metrics over skills independently. This would only be useful when the skills are known beforehand, and not discovered with data [8]. We now provide evidence that suggests that classification metrics may be misleading, even when they are not affected by the Simpson's paradox.

4.1.2 Do traditional metrics lead to good decisions?

We now compare Leopard and traditional metrics for choosing an item-to-skill mapping. We train a PFA model using our Math dataset. Table 2 compares the results of White ($R = 0.6$) and AUC.

If we were to choose the best skill mapping by AUC alone, we would choose the finer item-to-skill mapping, while White selects the coarser one. Why do they disagree? The fine skill mapping has almost three times the number of skills (90 skills) than the coarse mapping (27 skills). This means that for the effort to be the same on both models, the finer model should give a third of the practice of the coarser model. Even though the finer model is slightly more predictive, we argue that the coarser model is better suited for adaptive tutoring.

Table 2: Comparisons of item-to-skill definitions.

	White		
	score	effort	AUC
coarse	**.41**	**55.7**	.69
fine	.36	88.1	**.74**

4.1.3 Case Study

For completeness, Table 3 demonstrates using different student modeling techniques with the coarse item-to-skill mapping. For Knowledge Tracing, we show both the White estimates, and the Teal estimates (in parenthesis). We use the average sequence length for each skill because Teal requires a sequence length as an input. The estimates of Teal and White for effort are very similar, but their scores mismatch— possibly due to the differences in imputation for skills that don't reach the threshold. The low score metrics are indicative of students not reaching the performance threshold. This suggests that further inspection is necessary, because the learning curves may be decreasing or some some skills may have high slip probabilities. One of the advantages of White is that it can be used to evaluate non-probabilistic student models. For example, we use White to evaluate the student model that gives practice of a skill until the student gets three correct answers in the skill.

Table 3: Student model comparison using Leopard

	Leopard		
	score	effort	AUC
Knowledge Tracing	.39 (.18)	**49.5** (50.9)	**.70**
Performance Factor Analysis	.41	55.7	.69
Three Correct	.39	59.1	n/a
Majority Class	.41	65.6	.50

4.2 Simulations

With real data, we do not know the extent that the parameters are learned correctly, or affected by modeling problems— such as misspecification. We now use synthetic data to evaluate different metrics and compare them to a ground truth. Given that we know the Knowledge Tracing parameters that were used to generate the synthetic datasets, we can use Teal to calculate *exactly* the student effort and outcomes.

We sample 500 different datasets using random Knowledge Tracing parameters. In none of the datasets we allow forgetting, but we do not impose any other constraint (not even that students improve with practice). Each dataset has only a single skill, and has 200 students with 10 practice opportunities. We do not learn parameters from the synthetic

dataset, so we do not cross-validate.

4.2.1 *Which metrics correlate best with the truth?*

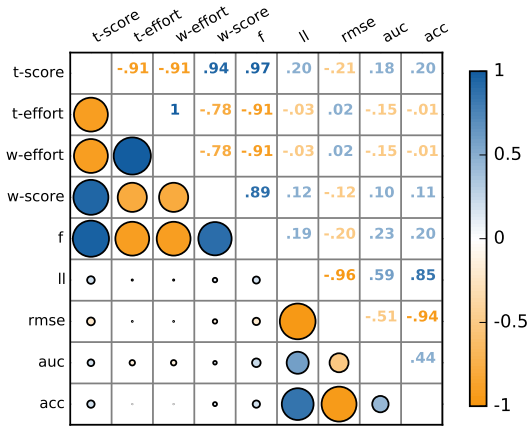


Figure 4: Correlation matrix of Leopard and conventional metrics. The size of the circles indicate the magnitude of the Pearson ρ correlation coefficient.

Figure 4 shows the pairwise Pearson-ρ correlations across 500 synthetic datasets on Teal (score), Teal (effort), White (effort), White (score), F-metric, Log-likelihood, RMSE, AUC, and Accuracy.

The metrics that correlate the most with the ground truth are White and the F-metric. Interestingly, the ground truth effort and score have low correlation with all the conventional metrics, except the F-metric, but the conventional metrics have relatively high correlation among each other (except the F-metric). In other words, most conventional metrics seem to be exchangeable.

We now investigate the effect of the imputation strategy of White. We are mindful that all of the synthetic students have 10 practice opportunities. Therefore, if White reports an effort of 10 for a dataset, it is likely that the dataset is not suitable for adaptivity, and that White may be imputing missing data to calculate the score. Figure 5 compares the 324 datasets that White reports effort lower than 9.99. Each dot in the scatterplot represents a different dataset. We see that effort computed with White has an almost perfect correlation with the ground truth ($\rho = 1.00$, $p<0.05$). On the other hand, the score computed with White is affected by our imputation strategy, but still has near perfect correlation ($\rho = 0.98$, $p<0.05$) with the ground truth. The correlation of the F-metric with the ground truth effort ($\rho = -0.47$) and score ($\rho = 0.89$) is relatively lower than White's. E.g., when the ground truth effort is 0, the F-metric ranges from very bad (0.2) to very good (1.0) predictive power, but White's effort is close to 0. Moreover, we speculate that score and effort may be more relatable to practitioners with little background of machine learning than the F-metric.

4.2.2 *Does White Converge to True Values?*

We now investigate whether White converges to the true values calculated by Teal. We use the same parameters used to plot Figure 2, and we manipulate the number of synthetic

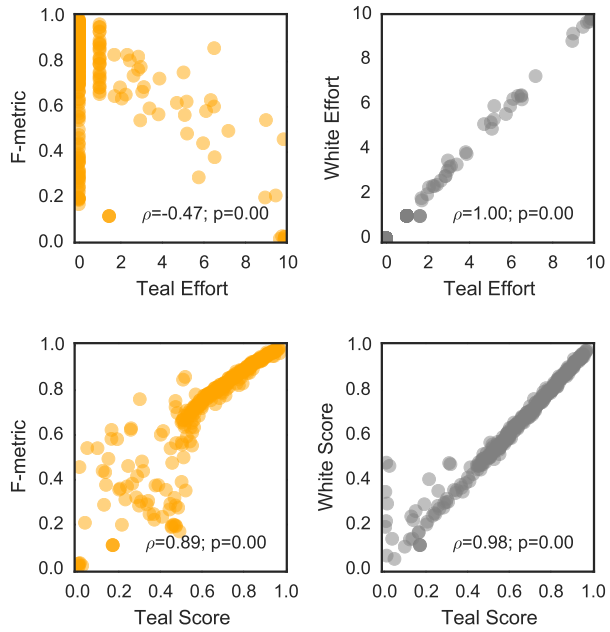


Figure 5: Comparison between F-metric and White to the ground truth.

students, each student with 20 practice opportunities, Figure 6 shows that with little data, White converges to the true value computed by Teal. Future work may provide a formal argument of when and how much data White requires to convergence.

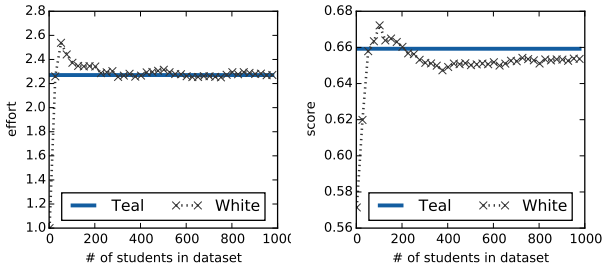


Figure 6: Example of White converging to Teal.

5. DISCUSSION

Our main contribution is the Leopard framework that automatically assesses adaptive tutoring systems in dimensions that relate to learner effort and outcomes. These dimensions were previously measured only in randomized control trials. We present Teal and White, two novel metrics that apply Leopard and are useful to evaluate adaptive tutoring systems. Secondary contributions include a novel methodology to assess evaluation metrics, the insight of Simpson's paradox affecting adaptive tutoring evaluation, and the implementation of the techniques we propose in this paper[3].

Classification evaluation metrics are very widespread in many disciplines, and their use in education is very important.

[3]`http://josepablogonzalez.com`

For example, for Computer-Adaptive Testing (CAT), classification metrics provide very useful insights to psychometric models. Leopard is not intended to replace classification metrics, randomized control trials, automatic experimentation [14], or visualization approaches [7, 11]. Leopard is a complementary approach to existing techniques, and we claim that it is specially useful when *in vivo* and online experimentation is not feasible.

We argue against the *de facto* standard of evaluating adaptive tutoring solely on classification metrics. Our experiments on real and synthetic data reveal that it is possible to have student models that are very predictive (as measured by traditional classification metrics), yet provide little to no value to the learner. Moreover, when we compare alternative tutoring systems with classification metrics, we discover that they may favor tutoring systems that require higher student effort with no evidence that students learn more. That is, when comparing two alternative systems, classification metrics may prefer a suboptimal system.

An interesting future direction may be to relax Teal's assumption that all sequences have fixed-length. Future work may provide more rigorous theoretical analysis on convergence, confidence intervals, validate our metrics with randomized control trials, or derive White for policies with multiple skills per item.

We are excited to see future work in adaptive tutoring systems reporting their contributions in terms of learner effort and outcomes. Besides the technical contributions of our evaluation metrics, we hope that our work contributes to the mission of driving the student modeling community to have a more learner-centric perspective.

6. REFERENCES

[1] R. Baker, A. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In B. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 406–415. Springer Berlin / Heidelberg, 2008.

[2] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In S. K. D'Mello, R. A. Calvo, and A. Olney, editors, *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013*, pages 4–11. International Educational Data Mining Society, 2013.

[3] B. S. Bloom. Learning for mastery. *Evaluation Comment*, 1(2):1–12, 1968.

[4] H. Cen, K. R. Koedinger, and B. Junker. Is Over Practice Necessary?—Improving Learning Efficiency with the Cognitive Tutor Through Educational Data Mining. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 511–518, Amsterdam, The Netherlands, 2007. IOS Press.

[5] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[6] A. Dhanani, S. Y. Lee, P. Phothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley, May 2014.

[7] I. M. Goldin and A. Galyardt. Viz-r: Using recency to improve student and domain models. In *Proceedings of the 2nd ACM conference on Learning At Scale*, Vancouver, Canada, Mar. 2015.

[8] J. P. González-Brenes. Modeling Skill Acquisition Over Time with Sequence and Topic Modeling. In G. Lebanon and S. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics AISTATS 2015*, pages 296–305, 2015.

[9] J. P. González-Brenes and Y. Huang. Using data from real and simulated learners to evaluate adaptive tutoring systems. In *Proceedings of the Workshops at the 18th International Conference on Artificial Intelligence in Education AIED 2015*, Madrid, Spain, 2015.

[10] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. In M. Mavrikis and B. M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, London, UK, 2014.

[11] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In J. G. Boticario, O. C. Santos, C. Romero, and M. Pechenizkiy, editors, *Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, Spain, 2015.

[12] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In K. Yacef, O. R. Zaïane, A. Hershkovitz, M. Yudelson, and J. C. Stamper, editors, *Proceedings of the 5th International Conference on Educational Data Mining*, pages 118–125, Chania, Greece, 2012.

[13] C. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51. Association for Computational Linguistics Morristown, NJ, USA, 2002.

[14] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popović. Towards automatic experimentation of educational knowledge. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3349–3358. ACM, 2014.

[15] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics Morristown, NJ, USA, 2001.

[16] Z. A. Pardos and M. V. Yudelson. Towards moment of learning accuracy. In *Simulated Learners Workshop of Artificial Intelligence in Education*, 2013.

[17] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.

[18] R. Pelánek. A Brief Overview of Metrics for Evaluation of Student Models . In S. Gutierrez-Santos and O. C. Santos, editors, *Approaching Twenty Years of Knowledge Tracing Workshop of the 7th International Conference on Educational Data Mining*, London, UK, 2014.

[19] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[20] D. Rai, Y. Gong, and J. E. Beck. Using dirichlet priors to improve model parameter plausibility. In T. Barnes, M. Desmarais, C. Romero, and S. Ventura, editors, *Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain, 2009.

[21] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.

[22] M. Walker, C. Kamm, and D. Litman. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3):363–377, 2001.

Automated Session-Quality Assessment for Human Tutoring Based on Expert Ratings of Tutoring Success

Benjamin D. Nye*
The University of Memphis
Memphis, TN 38152
benjamin.nye@gmail.com

Donald M. Morrison
The University of Memphis
Memphis, TN 38152
dmmrrson@memphis.edu

Borhan Samei
The University of Memphis
Memphis, TN 38152
bsamei@memphis.edu

ABSTRACT

Archived transcripts from tens of millions of online human tutoring sessions potentially contain important knowledge about how online tutors help, or fail to help, students learn. However, without ways of automatically analyzing these large corpora, any knowledge in this data will remain buried. One way to approach this issue is to train an estimator for the learning effectiveness of an online tutoring interaction. While significant work has been done on automated assessment of student responses and artifacts (e.g., essays), automated assessment has not traditionally automated assessments of human-to-human tutoring sessions. In this work, we trained a model for estimating tutoring session quality based on a corpus of 1438 online tutoring sessions rated by expert tutors. Each session was rated for evidence of learning (outcomes) and educational soundness (process). Session features for this model included dialog act classifications, mode classifications (e.g., Scaffolding), statistically distinctive subsequences of such classifications, dialog initiative (e.g., statements by tutor vs. student), and session length. The model correlated more highly with evidence of learning than educational soundness ratings, in part due to the greater difficulty of classifying tutoring modes. This model was then applied to a corpus of 242k online tutoring sessions, to examine the relationships between automated assessments and other available metadata (e.g., the tutor's self-assessment). On this large corpus, the automated assessments followed similar patterns as the expert rater's assessments, but with lower overall correlation strength. Based on the analyses presented, the assessment model for online tutoring sessions emulates the ratings of expert human tutors for session quality ratings with a reasonable degree of accuracy.

Keywords

Automated Assessment, Tutoring Dialog, Dialog Acts, Dialog Modes, Natural Language Processing, Educational Data Mining

*Corresponding Author

1. INTRODUCTION

As online learning has expanded, computer-mediated tutoring and help-seeking has become more prevalent and accessible. This tutoring occurs in a variety of forms, ranging from large commercial platforms employing certified teachers down to ad-hoc peer tutoring in rudimentary learning management systems (LMS). These systems generate a wealth of data about human tutoring interactions that can provide significant insights into the processes of online learning, the space of effective tutoring strategies, and the effectiveness of different platforms and contexts for tutoring. However, to study successful tutoring, tools are needed that can help distinguish between more and less successful sessions.

Quality ratings for tutoring sessions are often only available from self-reports by the tutor and student. However, these ratings have significant problems. Students typically have limited metacognitive skills and need training to assess their own learning [17]. Tutors can be more effective judges of learning, but a tutor's assessments of their students' learning can be biased and hard to compare due to these rating biases. Some of these biases may be individual variation (easy vs. hard raters), while others are systematic, such as less-expert tutors reporting higher average learning from their sessions. Other tutoring session sources have no real quality measure. For example, peer tutoring often lacks any assessment of the quality of the tutoring session, and hand-tagging these sessions for quality measures would be very time-consuming.

A standardized, automated estimator for the effectiveness of online tutoring sessions is arguably essential to the analysis of large-scale transcript corpora. Such a tool can be used to identify especially high-rated sessions, to track the results of improvement efforts, and to identify patterns in associated metadata. Also, differences between the automated estimator and tutors' self-reports could be used to identify new features that indicate effective tutoring strategies (i.e., an active learning approach). As such, the iterative improvement of a session success indicator would provide new insights into the features of effective tutoring and how they relate to other sets of data.

In this work, we have used a two-step supervised learning approach to train an estimator for session effectiveness. This

estimator was trained on a corpus of 1438 human-to-human tutoring sessions, where each session was rated in terms of two quality measures and each statement was annotated with a dialog act tag (e.g., *Confirmation:Positive*) and a dialog mode (e.g., Scaffolding). Based on the quality ratings assigned by independent expert tutors, features related to tutoring session success were identified using sequential pattern mining and statistical analysis of high-level session features (e.g., duration). Second, regression models that employed these features were trained to rate the quality of the tutoring sessions. Finally, this model was applied to a large sample of 246k tutoring sessions to examine the consistency of these ratings against metadata associated with each session, such as the original tutor's rating of student learning and of the student's knowledge of necessary prerequisites.

2. BACKGROUND AND RELATED WORK

Studying strategies and patterns in tutoring transcripts is a longstanding research area with roots in speech act theory [21]. Key techniques from this literature include dialog act classification [8], identifying dialog modes [1], and identifying statistically significant sequence patterns [3]. Our research described here relies on the use of all three levels of analysis to identify significant features that can be used to assess session quality. Dialog act classification involves binning each tutor or student statement into distinct taxonomy categories, which represent the functional purpose of the statement (e.g., an "Assertion" that states a fact). Dialog act taxonomy distinctions vary depending on the research focus, such as question types [8], higher-level dialog acts and feedback [1], and finer-grained pedagogical acts [3]. Our research extended this prior work in several ways, including a highly granular coding scheme, developed in collaboration with professional online tutors, which will be discussed later.

Dialog modes are a more recent area of focus for machine learning, but their theoretical underpinnings for studying learning are equally mature. In our work, modes represent shared understandings regarding hidden, higher-order dialog states with associated roles and expectations concerning the likelihood and appropriateness of particular dialog acts given that state [16]. In tutoring research, theoretically-based modes typically represent pedagogical strategies, such as Modeling, Scaffolding, and Fading. More recent studies of modes have used unsupervised approaches, such as Hidden Markov Models to detect patterns of dialog acts that match such theoretical modes [1]. However, such discovered states are not always guaranteed to be modes as we frame them here: others likely represent intermediate structures, such as adjacency pairs (e.g., a question followed by an answer). As such, in this research, we have relied on human-tagged modes and supervised mode-classifiers based on such modes, so that each mode can be linked more clearly to theoretical descriptions of pedagogy.

Finally, this research relies on features extracted using sequence data mining. A good review of prior work for sequence mining tutoring transcripts is presented by D'Mello and Graesser [3], which outlines conventional approaches (e.g., association rule mining) as well as a novel method based on transition likelihoods. In general, traditional analyses of tutoring sessions focus on identifying frequent or distinctive dialog act transitions and subsequences. However, where supervised labels exist (e.g., quality tags), alternative sequence analysis techniques can be applied to identify sequences that occur more frequently in certain session types. This type of analysis detects distinctive subsequences, which discriminate between one group of sequences versus another group of sequences [5].

Since online human tutoring is a dyadic interaction, it also has similarities with computer-supported collaborative learning (CSCL). CSCL analysis often considers higher-level constructs related to collaboration, such as reaching consensus and division of tasks [13]. Many of these constructs are less central to a professional tutoring process, which has predefined roles (tutor vs. student) and associated cultural expectations for dialog behavior. However, aspects of these more general interactions were incorporated, such as dialog management (a "Process Negotiation" mode) and interpersonal relationships (a "Rapport Building" mode).

The quality of a tutoring session can be measured in two ways: "objective" assessments, such as tests given to the student [1], or "subjective" assessments, based on expert ratings or tags assigned to the session. However, even objective assessments require subjective decisions about their criteria. Additionally, expert raters can often provide higher granularity for tagging events during the tutoring process. As such, process-focused machine learning often focuses on building classifiers and estimators trained on expert tags and ratings [18]. Our research builds on this approach, so our automated assessments model how expert tutors *perceive* session quality rather than necessarily the resulting learning gains. In future work, we feel that there would be great value in contrasting a session quality assessment trained on tested learning gains against the one developed in this paper. Such an assessment might identify session features that help identify when illusions of mastery and other rating biases occur [6].

3. DATA SET

This research analyzes a full data set of 246k online human-to-human tutoring transcripts from a major commercial tutoring service (Tutor.com). Thousands of different tutors, and tens of thousands of different students participated in these sessions, but all focused on Algebra and Physics topics. As an on-demand service, each session was initiated by a student who requested help on a problem or concept (e.g., at an impasse). Of these transcripts, approximately 4k were excluded from analyses on the full data set due to missing data or formatting issues. Each session contained a timestamped line-by-line text transcript of the statements typed by the student, the tutor, and system messages (e.g., file uploads). Every session was also associated with metadata collected before and after the session. This metadata included the tutor's assessment of evidence of learning during the session (EL1) and the tutor's assessment of the student's level of prerequisite knowledge (PREREQ). Metadata was also available for a subset of tutors, which included their "Tutor Level," an internal performance level that ranged from "Probationary" (0) to "Level III" (Highest). The tutor level was determined by each tutor's mentor, based on internal reviews of the tutor's sessions, and is correlated with experience. On average, Level III tutors had five years experience, Level II had two to four years, and Level I had a little over

a year. Probationary tutors averaged 6 months.

Of the total set of transcripts, 1438 sessions were annotated by a panel of 19 subject matter experts (SMEs), selected from a pool of some 2,800 Tutor.com tutors using a rigorous screening process, which included analysis of answers to a set of survey questions designed to gather initial expert opinion about tutoring, and also to assess the respondents' ability to critique session transcripts. The training process and details on inter-rater reliability are described in more detail in related work [15]. As part of the annotation process, the SMEs rated each session on two scales: evidence of learning (EL2) and educational soundness (ES). Annotators were instructed to consider different criteria for each: EL2 targets outcomes (i.e., did the student learn) and ES targets process (i.e., did the tutor use good tutoring strategies). This is important because sometimes good tutoring steps can still fail to produce learning for a given student. EL1, EL2, ES, and PREREQ were all rated on a 0-5 scale, where zero represents a low rating and five represents a top rating.

Each line in the tutoring session was also tagged for a dialog act and was also part of a dialog mode. Given the size of the taxonomies (126 dialog acts and 16 dialog modes), a full review of each tag would be infeasible, so specific tags that showed value as features will be noted as they are discussed. The taxonomy of dialog acts included 126 distinct tags, organized into 15 main categories. At a macro-level, these categories focus on traditional dialog act classes such as Questions, Assertions, Requests, Directives, and Expressives [21]. Within the tutoring context, these categories tend to be used to provide information (Answer, Assertion, Clarification, Confirmation, Correction, Expressive, Explanation, Reminder), asking for information (Hint, Prompt, Question), and managing the tutoring process (Directive, Promise, Request, Suggestion). Within each of the 15 main categories, subtypes capture key differences such as positive versus negative feedback (e.g., *Expressive:Positive* vs. *Expressive:Negative*).

Annotators also tagged student or tutor contributions that signaled the start of a dialog mode, or a switch from one dialog mode into another. The 16 included modes associated with classic tutoring strategies (Fading, Modeling, Scaffolding, Sensemaking, Session Summary, Telling), identifying the problem (Method Identification, Problem Identification) or learner prerequisites (Assessment), interpersonal strategies (Metacognitive Support, Rapport Building), and session process (Process Negotiation, Opening, Closing, Method Road Map, Off Topic). The time spent in each mode was far from uniform. Tutoring strategy modes, particularly Scaffolding, accounted for a majority of most sessions. Session process modes were also significant, such as Process Negotiation (i.e., getting on the same page), Openings, and Closings. Other modes were fairly rare, such as Method Identification.

Based on these annotated tags, complementary research on this data set developed a logistic regression dialog act classifier [20] and a conditional-random fields (CRF [11]) mode classifier [19]. This tagging methodology followed similar principles to Moldovan et al. [14]. These classifiers approached the level of reliability shown by independent tagging by human experts, as noted in Table 3. The figures in this table show the best performance by both the human taggers (i.e., their final inter-rater reliability tests) and the performance of the classifiers used for automated tagging in this paper. Machine tagging statistics shows cross-validation results. As can be observed, the classification of the main dialog acts (15 categories) and full set of sub-acts (126 categories) approximated human inter-rater tagging fairly closely. Classifying modes was fairly effective also, but lost nearly half of its accuracy the tagger trained on human speech act tags was applied to the machine-labeled dialog acts (29% accuracy). Retraining on machine tags before testing on machine tags improved overall accuracy, but still produced a significantly lower kappa (43% and 0.21, respectively, as shown in parentheses), as compared to training and testing on human tags. As such, mode tags will be less accurate for machine-tagged sessions.

Table 1: Reliability Scores for Tagging

	Main Act		Sub-Act		Mode	
Tagger	Acc	Kappa	Acc	Kappa	Acc	Kappa
Human	81%	0.77	65%	0.63	56%	0.47
Machine	77%	0.71	53%	0.50	57% (43%)	0.52 (0.21)

From the standpoint of analysis, the 1,438 human-tagged training set was used for initial feature identification and training of the session quality assessment model. The full set of 242k machine-tagged sessions were then treated as a second data sample for analysis, which included the original training set but tagged using the automated dialog act and mode classifier models. This research builds on the prior research that developed dialog act classifiers [20] and mode classifiers [19], as well as development of a taxonomy for speech acts and modes in human tutoring [15]. The novel contributions reported in this paper include identifying patterns in speech acts and modes (subsequence analysis), identifying features that help estimate tutoring session quality, training machine learning models that estimate tutoring session quality, examining the strength of features in these models, and examining the correlation between estimated session quality against other indicators of session quality (e.g., the original tutor's rating of learning during the session). This work was done to target the research questions described in the following section.

4. RESEARCH METHODOLOGY

Based on these data sets, this work approaches five primary research questions:

1. How closely can we model expert judgments about session quality, based on domain-independent dialog acts and modes?
2. What models show the most promise for assessing session quality?
3. What features are the strongest predictors in these models?
4. What features lose predictive power when trained on machine tags rather than human tags?

5. How closely do the results from machine quality tags correlate with metadata on the full corpus (e.g., EL1), as compared to the training corpus?

To examine these questions, a session quality classifier was trained using a two-step process of feature selection followed by supervised learning. First a set of high-level features was selected that correlated with the rater's evidence of learning (EL2) and educational soundness (ES). These features included the duration of the session, the average number of words typed by the student per contribution (verbosity), the number of dialog acts typed by the tutor and by the student, and the number of short and long pauses between dialog acts. Additionally, the counts of each mode tag and of each individual dialog act by a given speaker were used as features (e.g., *Confirmation:Positive [Tutor]*).

Next, to capture more complex features of the tutoring process, sequence pattern mining was applied to tutoring sessions to identify subsequences of dialog acts or dialog modes that help distinguish between excellent and poor tutoring sessions. For this analysis, two subsets of human-annotated tutoring sessions were selected that included the most successful sessions (N=261, where $ES = 5$ and $EL2 = 5$) and the least successful sessions (N=93, where $ES <= 2$ and $EL2 <= 2$). Subsequences of dialog modes consider dialog mode switches, where there was a change from one mode to another. This is important because modes often span multiple dialog acts.

The subsequence analysis used the TraMiner package for sequence analysis [5], which contains an algorithm for detecting discriminant event subsequences between two groups of sequences. At a high level, this algorithm calculates the frequency of all subsequences up to a given length for each group of sequences, then applies a Chi-squared test (Bonferroni-adjusted) to identify subsequences that are statistically more (or less) frequent in each group. In this context, a subsequence must be distinguished from a substring: subsequences are ordered, but do not necessarily have to be contiguous. Three sets of distinctive subsequence analyses were performed: 1) dialog act subsequences, 2) mode subsequences, and 3) dialog acts within each type of mode. Any subsequence which was distinctive at the $p<0.4$ level was included as a candidate feature. The $p<0.4$ cutoff was selected to allow a large set of candidate features, while still likely performing better than chance. This analysis was performed on the human-annotated tags. Each subsequence was treated as a feature whose incidence would be counted within a session (i.e., a count of the number of times that tags occurred in that order, without reusing any tags).

Four algorithms were trained to estimate the average of ES and EL2 based on the full feature set: linear regression with feature selection, support vector machine (SVM) regression [10], and additive regression based on decision stumps [4]. In general, these algorithms were selected and tuned to try to avoid over-fitting: the final number of active candidate features was 1465, which was comparable to the number of training sessions (1438). Ridge regression reduces the number of parameters by penalizing additional factors. Support Vector Machines are resistant to overfitting because they regularize the space solution space. Additive regression (also called Stochastic Gradient Boosting) uses smoothing that reduces the impact of each additional factor. Each algorithm was evaluated using 10-fold cross validation, using Weka [9]. After evaluating the effectiveness of each algorithm on the human-annotated data, the best of these algorithms was then tested on the machine-tagged sessions to examine performance. The best algorithm was re-trained using machine-tagged sessions, to test if calibrating to the dialog act and mode classifier outputs would improve performance.

Finally, the full set of 242k tutoring sessions was tagged using the best-fit model for session quality. These quality tags were correlated against session metadata available for the larger corpus of sessions: the original tutor's evidence of learning (EL1), the original tutor's assessment of the student's prerequisite knowledge (PREREQ), and the level of the tutor (Tutor Level). These correlations were compared against the correlations observed between the automated assessments and these same metadata variables for the training set. The goal of this analysis was to examine the consistency of the automated assessment with other ratings of session quality that were available for all tutoring sessions.

5. RESULTS AND DISCUSSION

The results from each step are discussed in this section, including sequence mining for session features, training and evaluating the session assessment model, and applying this model to a large corpus of online tutoring session transcripts. For the sake of brevity, dialog acts in this section are displayed using the shorthand form <Main Dialog Category>: <Sub Act> [<Speaker>], such that *Expr:Praise [T]* means "expressive praise from the tutor."

5.1 Sequence Pattern Mining

Discriminate sequence analysis that compared the most successful and least successful tutoring sessions identified 1151 better-than-chance ($p<0.4$) distinctive subsequences from 2 to 7 elements long. The majority of these sequences were sequences of dialog acts (1062) and a significant number of these sequences captured variations on similar patterns. Due to the granularity of the taxonomy, distinctions occurred such as *Assertion:Calculation [S]⇒ Expressive: Confirmation: Positive [T]* versus *Assertion:Calculation [S]⇒ Confirmation:Positive [T]*, where the only difference was whether the tutor's feedback took the form of an Expressive. Moreover, such distinctions sometimes showed slightly higher distinctiveness. For example, in the above case, *Expressive:Confirmation: Positive* feedback (e.g., *"Great!"*) was a stronger indicator of session success than *Confirmation: Positive* (e.g., *"Right"*).

A total of 89 distinctive mode subsequences were identified as candidate features that distinguished between session quality. Many of these were variants of eight patterns that were supported by Bonferroni-adjusted Chi-squared tests at the $p<0.05$ level. Six of these patterns were indicators of positive sessions. 1) Successful sessions almost always ended with a Closing/WrapUp, suggesting that both the tutor and student are satisfied with the progress. 2) Successful sessions had more Fading. The existence of even one Fading segment was an indicator of success, though Scaffolding preceding Fading was a better indicator; 3) Successful ses-

sions tended to have repeated Scaffolding or Sensemaking segments (the conceptual equivalent of Scaffolding), where Scaffolding was interleaved with other modes. 4) Successful sessions were more likely to have late-session Rapport Building is after Scaffolding or Fading, but preceding the Closing. 5) A Telling mode (i.e., mini-lecture) before Rapport Building was also a positive feature, which likely indicates that a summary is positive. 6) The presence of a single Opening mode was also an indicator of a good session, where less-successful sessions skipped the Opening greetings and moved immediately to Problem Identification.

Two patterns of mode subsequences tended to be associated with less successful tutoring sessions. 1) Unsuccessful sessions tended to have repeated Modeling mode cycles. While a single Modeling mode segment was not indicative of a poor session, two or more in series was associated with worse ratings. 2) Unsuccessful sessions were also indicated by repeated Process Negotiation, particularly if Process Negotiation alternated with Modeling (the tutor solving the problem) or Problem Identification (figuring out what problem the student has). It was also a negative indicator when Process Negotiation started early in a session sequence. Process Negotiation is a mode that is associated with discussing the tutoring process itself, which includes figuring out who should be speaking or addressing technical issues. Process Negotiation itself was not a bad mode, and was also present in many good characteristic sequences. In these good sequences, it tends to occur late in the session (preceding a Closing) rather than early-on. In general, long or early cycles of Process Negotiation likely indicate that the student is unable to contribute meaningfully to the problem due to lack of prerequisites, technical issues, or poor dialog coordination (e.g., student interrupting).

From aligning these distinctive subsequences, an ideal path of modes for a session might be framed as: Opening ⇒ ProblemID ⇒ Scaffolding ⇒ Fading⇒ ProcessNegotiation ⇒ Telling ⇒ RapportBuilding ⇒ Closing, where some modes (e.g., Scaffolding and Fading) optimally alternate multiple times. This successful mode sequence shows some similarities and differences when compared to Graesser et al.'s 5-step frame for in-person tutoring, which can be described as: [Tutor poses a question]⇒ [Student attempts to answer]⇒ [Tutor provides brief feedback]⇒ [Collaborative interaction]⇒ [Tutor checks if student understands] [7]. The final two frames align well with Scaffolding⇒ Fading⇒ ProcessNegotiation pattern observed in the successful online sessions. The main differences likely stem from the tutoring context. The Graesser tutoring frame assumes a tutor-driven process in which the student is attempting to answer a question, typically conceptual, posed by the tutor. In our data, the student is typically coming to the tutor for help on a specific problem, and the session is in this sense student-driven. As such, Problem Identification occurs first, instead of the tutor posing an initial question.

The insights from the dialog act sequences for successful versus less successful sessions show similar patterns as those based on sequences of modes. However, they are more granular and some of the distinctive sequences tend to be longer or repeating (e.g., repeated answers by a student alternating with *Confirmation:Positive* by the tutor are better). These patterns match loosely to the learning-relevant affective states noted by D'Mello and Graesser [2], which were: Achievement, Engagement, Disengagement, Confusion / Uncertainty, and Frustration. Evidence of achievement (i.e., answers that received positive feedback, explanations followed by expressions of understanding) corresponded with higher session ratings. Likewise, engagement (student answer attempts and sequences with multiple student statements) were positive.

Disengagement indicators, such as questions followed by *Expressive:LineCheck* (e.g., *"Are you there?"*) and *Expressive:Neutral* statements by the student (e.g., *"ok"*) were associated with lower ratings. Raters likely interpreted neutral responses as indicating that the learner was passively processing the session. By comparison, tutor questions that transitioned to *Confirmation:Understanding:Negative* (e.g., *"No, I don't understand"*) were not strong indicators of an unsuccessful session. Frustration was not significantly observed in the corpus, in part due to a lack of taxonomy tags devoted to detecting it and in part due to a relatively low prevalence of obvious frustration within the training corpus. While taxonomy acts for confusion and uncertainty were available in the taxonomy, these were less common and did not have a clear correlation to successful or unsuccessful sessions. This is somewhat expected, since a limited amount of confusion tends to be productive [2], but a large amount can lead to unproductive frustration. More nuanced techniques might be needed to monitor these cycles in tutoring sessions.

5.2 Automated Assessment Models

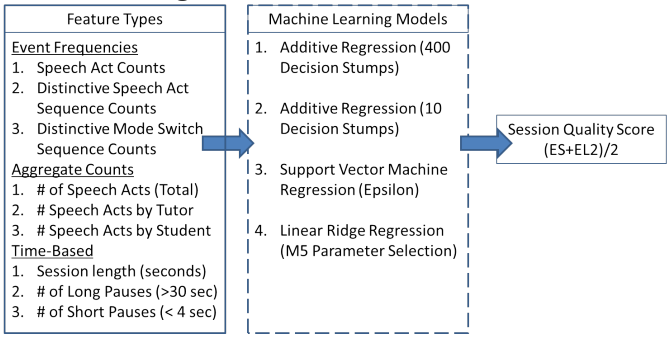
The total feature set was used to train a series of machine-learning models: linear ridge regression with parameter selection (Linear), SVM regression (SVM), and additive regression with decision stumps (Add.). The outcome variable for this training was a unified quality score based on the average of the rater's assessment of educational soundness (ES) and evidence of learning (EL2). The process for training these models is outlined in Figure 1. The results of 10-fold cross-validation for the best-fit models are presented in Table 5.2, in terms of the correlations between the machine-generated tags and the hold-out folds. Additive regression outperformed the other models, even with a fairly small number of decision branches (10). However, it improved significantly when allowed to use additional decisions (400). From examining the decision stumps, these additional stumps allowed it to incorporate additional factors and also form piecewise curves for some of the strongest factors.

Table 2: Regression Fits for (ES+EL2)/2 (10-fold CV)

	Linear	SVM	Add. (10)	Add. (400)
Human Tags	0.24	0.55	0.62	0.69
Machine Tags	0.24	0.49	0.52	0.56

The linear model performed very badly, despite parameter selection: it tended to overfit the data and did not seem to model the expert ratings very well. SVM performed slightly better, but was not the best model overall. The Additive model, which was based on decision thresholds, worked best

Figure 1: Model Data Flow



out of the three. This may indicate that the human raters tended to implicitly use heuristics such as "too many Modeling modes," or "not enough Student contributions." The nature of features was also a factor, since many features were relatively sparse in each session (e.g., only occurred once or twice within an average session), which lends itself to rules related to the existence of a feature (i.e., N > 0).

Models trained on the machine-generated tags followed a similar pattern, but with slightly worse estimates. Retraining the classifiers on the machine-labeled tags did not significantly improve estimates based on those tags. When applying the model trained on human tags to the training set with machine tags, the model fit is R=0.54, as compared to R=0.56 for the cross-validated model built on the machine tags. As such, the machine tags appear to lose certain information, rather than simply categorizing it differently.

Since the smallest Additive Regression model worked so effectively, it is worthwhile to examine the features that were included. These models differed slightly when trained on human tags versus machine-labeled tags. The top features for this model on human tags vs. machine-labeled tags are shown in Table 5.2, in order of their importance (note: *Confirmation* is shortened to *Conf*). The presented analysis used non-standardized data, which is reasonable partly because the length of Tutor.com sessions tends to be fairly regular (i.e., a typical session is 15-25 minutes). Normalization would likely be needed to apply this to significantly different corpora. In general, many of the same patterns are important for both the human and machine tagged models. At least some of the judgments are based on a required minimal session length (e.g., # of Tutor Acts). Certain features appear to target evidence of learning (EL2), such as tutor actions that indicate the student has provided correct answers (*Confirmation:Positive*,*Expr:Praise*) and not passive in the tutoring session (*Expr:Neutral*, *Expr:LineCheck*). Other features appear to be associated with educational soundness (ES) for tutoring process (e.g., existence of a Closing, Scaffolding, and no excessive Modeling). Machine tagging appears to lose some of these nuances with respect to modes, probably due to the significantly lower accuracy for classifying modes.

Overall, the model appears to capture evidence of learning (EL2) better than educational soundness (ES). When trained on the full training data set (human tags), the Additive Regression (400) correlates with the average of ES and EL2 at R=0.8. By comparison, the correlation to these estimates is R=0.76 for EL2 versus R=0.63 for ES. Clearly, this is not the result of the outcome variable itself, which is a straight average of the two ratings (R=0.93 with EL2 and R=0.92 with ES). Instead, this indicates that the features for evidence of learning are more easily detected using the available taxonomy tags and features. This limitation was amplified when using the machine-generated tags, where the fit to (ES+EL2)/2 was R=0.54 but the correlation with the components was R=0.55 for ES2 and R=0.38 for ES. As such, improving the automated tagging of dialog modes would improve the automated assessments significantly.

Table 3: Top-10 Features in Additive Regression

Trained on Human Tags	Trained on Machine Tags
Closing > 0	# of Tutor Acts > 11
Expr:Conf:Positive [T]⇒ *Expr:Conf: Positive [T]* > 0	RapportBuild ⇒ Closing > 0
Scaffolding > 0	*Expr:Conf:Positive [T]*⇒ *Expr:Conf: Positive [T]* > 0
Closing > 0	*Assertion:Concept [T]* < 18
Expr:Apology [T] = 0	# of Tutor Acts < 12
# of Tutor Acts > 6	# of Tutor Acts > 5
ProcessNegotiation⇒ Modeling ⇒ Modeling⇒ Modeling < 4	*Request:Conf: Understanding [S]* < 3
Expr:Praise [T] > 0	Scaffolding⇒ Scaffolding⇒ Closing > 4
Expr:LineCheck [T] = 0	# of Tutor Acts < 12
Expr:Neutral [S] > 15	*Expr:Conf:Positive [S]* > 1

5.3 Tagging Large Tutoring Data Set

To examine the consistency of this assessment model on out of sample data, it was applied to a corpus of 242k machine-tagged sessions. The features for each tutoring session were extracted from parsing the transcript. Metadata about the session and the tutor were collected and aligned to the automated session assessments for analysis. The correlations between the Automated Estimates (Estimates), EL1, and PREREQ were available for almost the full corpus of 242k sessions. Other metadata was not always complete (e.g., not all tutor level data was available), so each pairwise correlation may have a slightly different N. However, all comparisons involve thousands of values and are statistically significant at the p<0.01 level.

Table 4: Correlations of Quality Scores with Session Metadata

	Estimate	(ES+EL)/2	EL1	PREREQ
(ES+EL)/2	0.54	-	-	-
EL1	0.45	0.56	-	-
PREREQ	0.39	0.49	0.87	-
Tutor Level	0.05	0.11	-0.02	-0.04

Table 5.3 shows the correlations between the automated estimate of session quality (Estimate), the average quality score for human raters (ES+EL2)/2 (available for the training set only), the original tutor's ratings for evidence of learning (EL1) and the learner's prerequisite knowledge (PREREQ), and the Tutor Level. The first two columns of this ta-

ble show that the estimate maintains similar correlations to those for the ratings that it was based on, across the larger data set, but slightly weaker overall. For example, the session tutor's rating of learning for the student correlates at R=0.56 (N=1438) for the training tags, but only R=0.45 (N=242k) for the automated tags across the full session data. With that said, the automated session rating maintains a similar pattern as the supervised tags across the full corpus. This indicates that the automated assessment captures significant information from the original expert raters, but with additional noise due to the machine-tagging process (particularly for modes).

This table also indicates why an external rating source can be important for evaluating the quality of tutoring sessions, even for well-trained professional tutors. Despite being rated independently by tutors with no knowledge of the original tutor, a higher Tutor Level correlated with significantly higher external quality ratings (R=0.11, N=1328). However, these more-expert tutors rated both the learning (R=-0.02) and the prerequisite knowledge (R=-0.04) lower than lower-level tutors. Or, put another way, less-expert tutors probably over-estimate both the learning and initial understanding of their students.

Moreover, it may be difficult for session tutors to provide ratings for the session that capture distinct features. For example, the original tutors expressed an R=0.87 (N=242k) correlation between learning (EL1) and and prerequisite knowledge (PREREQ). While one would expect these factors to be related, that level of correlation is nearly identical. By comparison, the external quality ratings correlated with the PREREQ assessments much more loosely (R=0.49, N=1438) and the automated assessments shadow this pattern (R=0.39, N=242k). So then, this automated rater provides a unique source of information modeled after the judgments of the external raters, which can be complementary to other sources of information about tutoring session quality.

6. CONCLUSIONS AND FUTURE WORK

This research has offered some insights into the five primary research questions posed earlier in Section 4. First, this work demonstrates the feasibility of an automated assessment model that models human expert judgments about the learning that took place during an online human-to-human tutoring session, at a level of R=0.54. While room for improvement exists, this model is already functionally useful. At least in this work, non-linear meta-models based on decision stumps (e.g., Additive Regression) outperformed more linear approaches such as Linear Regression and SVM Regression. This finding indicates that Random Forests [12] and similar algorithms are probably also promising for this type of problem. The strongest predictors of session quality in these models tended to be features where the tutor confirmed the accuracy of the student's responses, the session process indicated that progress was occurring (e.g., Scaffolding, Fading), or a consensus about successful learning was reached (i.e., a mutually-agreed Closing). Of these features, modes were fragile when machine tags were used: the level of noise in the mode classification appears to wash out information that is needed to evaluate the tutoring process. Finally, the resulting model was shown to follow similar patterns to the original training ratings, even over a much larger data set. This indicates that the automated assessments offer a reasonable proxy for expert human assessment when needed.

Notably, these ratings are calculated without a domain model that can directly assess the quality of students' answers. Instead, the model captures more general features of the tutoring interaction that relate to engagement and consensus between the tutor and student about learning accomplishment. As such, this model should be effective across a variety of tutoring domains beyond those analyzed in this work (Algebra and Physics). These session features are, in principle, domain-independent: they are based on classifications of tutoring dialog acts and modes.

However, this is also a limitation. Since the automated assessment system lacks the ability to assess the correctness of student input, it relies significantly on the session tutor's domain knowledge and basic capabilities to provide correctness feedback. As such, the session assessments can detect aspects of the pedagogy and student progress, but are unlikely to work appropriately if the tutors are entirely unqualified. This is, in part, because the training corpus includes only professional tutors who are rated and evaluated for quality. As such, additional quality-rated corpora might be needed to transition this estimator to other tutoring contexts where session quality assessments are important (e.g., peer-tutoring).

Additionally, significant drops in performance were observed when using machine-annotated sessions instead of human-annotated sessions. These drops were particularly severe for mode classifications, which had a direct impact on the ability of the session quality estimates to model the educational soundness of a session. This functionality would be helpful, as it allows credit for "good process" even when strong learning outcomes are not observed. Improving the accuracy of dialog mode classification would significantly strengthen the assessment of tutoring sessions, and is an important area for further research. One way to approach this problem would be to use active learning where machine-annotated transcripts are corrected by human taggers.

Finally, an important next direction for this research would be to train a similar tutoring session assessment model based on pre-test and post-test assessments, such as the approach taken by Boyer et al. [1]. This step would enable a comparison between the features underlying our expert ratings of session quality against the features associated with measured learning gains. This work may show notable qualitative differences related to not only the key features, but also the algorithms involved (e.g., discontinuous algorithms such as Additive Regression might not be as dominant). Features associated with learning gains that are not associated with human ratings might also help detect illusions of mastery or expert blind spots. Likewise, integrating both approaches for analysis of tutoring sessions would offer the potential to identify authentic "Eureka moments" where the learner's sense of sudden understanding can be shown to correlate with subsequent performance on a similar problem. In the long term, the process of maintaining and improving this model should provide insights into new features of successful tutoring that may even be more valuable than the automated assessments calculated by the model.

7. REFERENCES

[1] K. E. Boyer, R. Phillips, A. Ingram, E. Y. Ha, M. Wallis, M. Vouk, and J. Leste. Investigating the relationship between dialogue structure and tutoring effectiveness: a hidden markov modeling approach. *International Journal of AI in Education*, 21(1-2):65–81, Jan. 2011.

[2] S. D'Mello and A. Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, 2012.

[3] S. D'Mello, A. Olney, and N. Person. Mining collaborative patterns in tutorial dialogues, Dec. 2010.

[4] J. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378, 2002.

[5] A. Gabadinho, G. Ritschard, N. S. Muller, and M. Studer. Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software*, 40(4):1–37, 2011.

[6] A. Graesser, S. D'Mello, and W. Cade. Instruction based on tutoring. In *Handbook of Research on Learning*, pages 408–426. 2011.

[7] A. Graesser and N. K. Person. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9(6):495—-522, 1995.

[8] A. C. Graesser and N. K. Person. Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137, Jan. 1994.

[9] M. Hall, E. Frank, and G. Holmes. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[10] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18—-28, 1998.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.

[12] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[13] A. Meier, H. Spada, and N. Rummel. A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning*, 2(1):63–86, Feb. 2007.

[14] C. Moldovan, V. Rus, and A. Graesser. Automated speech act classification for online chat. In *Midwest Artificial Intelligence and Cognitive Science Conference*, pages 23–29, 2011.

[15] D. Morrison, B. D. Nye, and V. Rus. Tutorial dialogue modes in a large corpus of online tutoring transcripts. In *Artificial Intelligence in Education (AIED) 2015*, Under review.

[16] D. Morrison and V. Rus. Defining the nature of human pedagogical interaction. In R. A. Sottilare, X. Hu, H. Holden, and K. Brawner, editors, *Generalized Intelligent Framework for Tutoring Systems, Volume 2: Pedagogical Strategies*, pages 217–224. 2014.

[17] I. Roll, V. Aleven, B. M. McLaren, and K. R. Koedinger. Metacognitive practice makes perfect: Improving students' self-assessment skills with an intelligent tutoring system. In A. Biswas, G and Bull, S and Kay, J and Mitrovic, editor, *AIED 2011*, volume 6738 of *LNAI*, pages 288–295, 2011.

[18] C. Rosé, Y.-C. Wang, Y. Cui, J. Arguello, K. Stegmann, A. Weinberger, and F. Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 3(3):237–271, Jan. 2008.

[19] V. Rus and N. Niraula. Automated labeling of dialogue modes in tutorial dialogues,. Technical report, The Language and Information Processing Lab, The University of Memphis, 2014.

[20] B. Samei, V. Rus, B. D. Nye, and D. M. Morrison. Hierarchical dialogue act classification in online tutoring sessions. In *Educational Data Mining (EDM) 2015*, In Press.

[21] J. Searle, F. Kiefer, and M. Bierwisch. *Speech act theory and pragmatics*. 1980.

A Framework for Multifaceted Evaluation of Student Models

Yun Huang
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
yuh43@pitt.edu

José P. González-Brenes
Pearson Research &
Innovation Network
Philadelphia, PA, USA
jose.gonzalez-brenes@pearson.com

Rohit Kumar
Speech, Language and Multimedia
Raytheon BBN Technologies
Cambridge, MA, USA
rkumar@bbn.com

Peter Brusilovsky
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA, USA
peterb@pitt.edu

ABSTRACT

Latent variable models, such as the popular Knowledge Tracing method, are often used to enable adaptive tutoring systems to personalize education. However, finding optimal model parameters is usually a difficult non-convex optimization problem when considering latent variable models. Prior work has reported that latent variable models obtained from educational data vary in their predictive performance, plausibility, and consistency. Unfortunately, there are still no unified quantitative measurements of these properties. This paper suggests a general unified framework (that we call Polygon) for multifaceted evaluation of student models. The framework takes all three dimensions mentioned above into consideration and offers novel metrics for the quantitative comparison of different student models. These properties affect the effectiveness of the tutoring experience in a way that traditional predictive performance metrics fall short. The present work demonstrates our methodology of comparing Knowledge Tracing with a recent model called Feature-Aware Student Knowledge Tracing (FAST) on datasets from different tutoring systems. Our analysis suggests that FAST generally improves on Knowledge Tracing along all dimensions studied.

Keywords

Student Modeling, Knowledge Tracing, parameter estimation, Identifiability, Model Degeneracy

1. INTRODUCTION

Adaptive tutoring systems often rely on student models to trace the progress of student knowledge to personalize instruction. Such student models are usually latent variable models with the state of student knowledge as the latent variable. However, finding optimal model parameters is usually a difficult non-convex optimization problem for latent variable models. Moreover, in the context of tutoring systems, even global optimum model parameters may not be interpretable (or plausible). Knowledge Tracing [4] is one such latent variable model that has been widely used, and different properties of its estimated parameters have been presented in many previous studies: predictive performance [6], plausibility [1, 6, 19], and consistency [2, 6, 16, 19, 9]. Unfortunately, there are still no unified quantitative measurements of these properties. If prediction of student performance is our only goal, this need is less urgent, since we can simply pick a model according to classification metrics. However, parameters with varying properties might have different inferences about knowledge, which may result in different tutoring decisions that can have a large impact on students. To illustrate, we show examples where two models that both belong to Knowledge Tracing are fitted from the same data, and where predictive performance is not sufficient to pick a good model:

- One model with higher predictive performance asserts that student knowledge decreases with correct practices, while the other model asserts the opposite. In such cases, the former model will suggest continuing practicing even if students get a lot of correct answers in a row, while the latter will suggest moving to other skills in a shorter amount of time.

- Two models have the same predictive performance, yet one asserts that about 20 practices are required to reach mastery of a skill, while the other asserts that only about 3 practices are enough. In such cases, a student needs to practice a lot under the former model, but under the latter model, students can move to learning other skills more quickly.

In the first example, the more predictive model lacks plausibility; in the second example, two models lack consistency, even though they have the same predictive performance. As a result, we advocate that a student model should be examined from dimensions besides predictive performance. We propose a unified quantitative framework, called Polygon, for the multifaceted evaluation and comparison of student models. The framework suggests novel metrics to quantify the properties of a student model along multiple dimensions, including predictive performance, plausibility, and consistency. Polygon is designed for general latent variable models that model latent student knowledge and is domain-independent. In the present work, we demonstrate how we apply Polygon to evaluate and compare classic Knowledge Tracing with a recent generalized model called Feature-Aware Student Knowledge Tracing (FAST) [8] in four different domains. Section 2 reviews some latent variable student models and prior work examining their properties; Section 3 describes our Polygon framework and metrics; Section 4 studies the relationship among these metrics and compares Knowledge Tracing with FAST; Section 5 concludes the work.

2. BACKGROUND

2.1 Latent Variable Student Models

We now review two effective latent variable models for predicting student performance and inferring student knowledge: Knowledge Tracing [4] and Feature-Aware Student Knowledge Tracing (FAST) [8]. Knowledge Tracing uses Hidden Markov Models to model student knowledge as binary latent variables (either learned or unlearned), given the observed practice performance (correct or incorrect) and using four parameters: Init (initial knowledge level), Learn (learning rate), Guess, and Slip. We learn the parameters of Knowledge Tracing using the Expectation Maximization algorithm. A recent model FAST incorporates features into Knowledge Tracing by replacing the binomial distributions by logistic regression distributions. It encodes contextual information as features for the original Knowledge Tracing parameters. It allows flexible features to affect student performance or knowledge directly. For simplicity, we use features in all four parameters in the study. FAST trains feature coefficients jointly with other parameters using the Expectation Maximization with Features algorithm [3]. This algorithm keeps the original E-step and replaces the M-step by training a weighted regularized logistic regression using a gradient-based search algorithm (LBFGS). While FAST has been shown to outperform Knowledge Tracing in many prediction tasks, we are interested in comparing it with Knowledge Tracing in other dimensions.

2.2 Prior Work Examining Properties of Knowledge Tracing

Prior work has examined Knowledge Tracing models from predictive performance, plausibility, and consistency. We now review previous studies in each dimension.

Predictive Performance. Measurements of predictive performance have been broadly applied to evaluate student models. Prior studies have shown several problems with parameter estimation for Knowledge Tracing, which predictive performance metrics often fail to detect [2, 16, 7]. We examine this traditional dimension in more depth for both Knowledge Tracing and FAST, and complement it in other dimensions, including plausibility and consistency.

Plausibility. Interpretability of a model is a desire property because it allows for better scientific claims and practical applications. Prior studies have used external measurements for validating the plausibility of fitted parameters, such as pre-test scores [6], exercise scores [4], or some domain-specific measurements [2]. However, such external resources are not always available. Many studies also examined plausibility by internal validity. Learning curves plotted using fitted parameters are inspected [2], and extremely low learning rates are considered implausible. However, very difficult skills can have very low learning rates, and it is not clear what is the suitable threshold for defining low learning rates. Implausibility has been formally defined using model degeneracy [1], which refers to situations where parameter values violate the model's conceptual meaning. They defined strong empirical constraints to detect theoretical degeneracy, and designed two specific metrics involving empirical parameters to detect empirical degeneracy: (i) the model's estimated probability that a student knows a skill is not higher than before the student's first N actions, or (ii) the model doesn't assess that the student has mastered the skill, even though the student has made a large number M of correct responses in a row. Under these two cases, the model is judged to be empirically degenerate. They arbitrarily chose N=3 and M=10 for the study. A later theoretical fixed point analysis [19] has precisely identified the conditions where models will be empirically degenerate. We are interested in generally quantifying the plausibility property based on such a theoretical conclusion, avoiding imposing empirical parameters during evaluation.

Consistency. Prior work has focused on two aspects of this dimension. First, the optimization algorithm (namely, the Expectation Maximization algorithm) can converge to the local optima of the log likelihood space yielding different properties of parameters that depend on the initial values [5, 16]. Although there are studies on setting good initial values to tackle this problem [5], practically, the strategy of setting randomly distributed initial values is usually taken. Yet there is still no principled way to measure the models' difference in the variation of convergence, and as a result, it is difficult to get a quantitative view of such a property. Second, multiple global optima of Knowledge Tracing exist [16, 2] where observed student performance corresponds to different sets of parameter estimates that make different assertions about student knowledge, yet have identical (under finite precision) performance predictions [2]. This problem is referred to as the identifiability problem [2]. Later studies have presented different (and even contradictory) views of this problem [19, 9]. These two aspects all relate to the consistency of the parameter space, and in order to determine their practical implications, we offer a unified view of them.

3. POLYGON EVALUATION FRAMEWORK

Polygon is a novel framework proposed for evaluating general latent variable student models from multiple dimensions with multiple metrics, besides simply predictive performance. It considers three dimensions, predictive performance, plausibility, and consistency, along with novel met-

rics that instantiate each dimension. Polygon can evaluate a single model which contains only one set of parameters fitted from the data, because in practice we usually deploy a single model into a tutoring system after model selection. Polygon's predictive performance and plausibility metrics can be used to evaluate single models. However, latent variable models can converge to different points with different initial parameter values due to the non-convexity of the negative log-likelihood. A better model should be more likely to converge to points with higher predictive performance and plausibility, and also give more stable predictions and inferences. So we also use Polygon to evaluate a student model fitted from a large number of random initializations. This provides an examination on the parameter space that is useful for single model selection or construction. In our study, we call these final fitted models random restarts. We mainly focus on evaluating the parameter space from random restarts, but also include evaluating a single model. Each Polygon metric evaluates the trained model(s) of a skill. To get an overall evaluation across skills, we aggregate by averaging each skill's individual metric. All metrics range from 0 to 1, with a higher positive value indicating higher quality. We focus on the evaluation on Knowledge Tracing and FAST in this study. We now introduce Polygon in detail.

3.1 Predictive Performance

Predictive performance has been the previous standard of evaluating student models. It provides useful validation for the inference of knowledge, since accurate knowledge estimation should imply accurate prediction of student performance. We apply a widely used classification metric for this.

AUC and P-RAUC. We use Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to evaluate each single model on test set, which gives an overall summary of diagnostic accuracy. AUC equals 0.5 for a random classifier and 1.0 for perfect accuracy. For assessing multiple random restarts, we compute the average of AUC values from single models and define it as P-RAUC, where P- stands for prediction performance, R stands for random restart, and r indicates the r^{th} random restart:

$$\text{P-RAUC} = \frac{1}{R}\sum_{r=1}^{R}\text{AUC}^r \tag{1}$$

3.2 Plausibility

The conceptual idea behind using Knowledge Tracing to model student knowledge is that knowing a skill generally leads to correct performance, and conversely, that correct performance implies that a student knows the relevant skill [1]. We define plausibility metrics based on this idea.

Guess+Slip<1 (GS) and P-RGS. Several prior studies have empirically addressed the issue of plausibility, as mentioned in Section 2. A recent study [19] has provided a theoretical ground that we think can be used to formally define plausibility. This study used theoretical fixed point analysis to prove that when Guess+Slip>1, the probability that a student has learned a skill just after a practice, given the student's previous performance, decreases for correct practices and increases for incorrect practices. In this case, the model is empirically degenerate [1]. This is different from theoretically degenerate [1] constraining Guess$\leq$0.5 and Slip$\leq$0.5 to be plausible estimations, which we think is somewhat too strong. For example, it is possible that a student may answer a problem correctly after receiving strong scaffolding (help), even though the skill has not yet been learned. As a result, we propose a metric constructed using the Guess+Slip<1 condition. We use an indicator for Guess+Slip<1 for a single model and refer to it as GS (Equation 2). For assessing random restarts, we compute the average of the GS values from single models and define it as P-RGS, where P- stands for plausibility and R stands for random restart (Equation 3):

$$\text{GS}^r = \mathbb{1}(\text{Guess}^r + \text{Slip}^r < 1) \tag{2}$$

$$\text{P-RGS} = \frac{1}{R}\sum_{r=1}^{R}\text{GS}^r \tag{3}$$

Here, $\mathbb{1}$ is an indicator function and Guess^r and Slip^r are the r^{th} random restart's fitted probabilities. For FAST, with the change of feature values, Guess and Slip can change. We focus on capturing the average behavior of guessing and slipping across contexts, so we compute Guess and Slip with only the intercepts in the logistic regression component (note that other features are activated according to context during training). The interpretation of our computation depends on the construction of features. For example, when using item indicator features, the computation captures the average values of Guess and Slip of a skill.

Non-decreasing Predicted Probability of Learned (NPL) and P-RNPL. In addition to the above metric grounded in a theoretical analysis [19] for Knowledge Tracing, we construct another empirical metric to capture the behavior of a general latent variable model, since it is not always easy or feasible to conduct theoretical analysis of complex models. Our proposed metric captures how likely a model gives a non-decreasing estimation of knowledge levels with an increase in practice opportunities. This idea is consistent with constraining the learning rate to be non-negative, as in [17, 6]. We think that a decreasing predicted probability of learned is not plausible, based on the interpretation that such a decrease implies practices that hurt learning. We are aware that a decreasing knowledge estimate can also be interpreted as a decrease in the model's belief that a student might reach a high knowledge level, where the model adjusts itself when observing a lot of incorrect practices. However, we focus on the first interpretation, because in real world tutoring systems where students are aware of their knowledge level as provided by the systems, decreasing knowledge estimates with more practices might discourage students from trying more.

To construct this new metric, we first obtain the estimation of a student reaching leaned state at each t^{th} practice opportunity given prior 1^{th} to $(t-1)^{th}$ performance O_1 to O_{t-1} on the test set. We denote this probability as $\text{P}(\text{L}_t{=}\text{Learned}|\mathbf{O}_{1:t-1})$, and also refer to it as $\text{P}(\widetilde{L}_t|\mathbf{O})$ for simplicity. Then we count the total number of consecutive pairs with non-decreasing $\text{P}(\widetilde{L}_t|\mathbf{O})$ across each skill-student sequence, and then divide it by the the total number of observations of the current skill. We define this as NPL as an indicator of its plausibility for assessing a single model (Equation 4). For assessing random restarts, we compute the average of the NPL values obtained from single models, and define it as P-RNPL, where P- stands for plausibility

and R stands for random restart (Equation 5):

$$\text{NPL}^r = \frac{1}{D}\sum_{s=1}^{S}\sum_{t=1}^{T_s-1}\mathbb{1}[\text{P}(\widetilde{L}_{t+1}^{rs}|\mathbf{O}^{rs}) \geq \text{P}(\widetilde{L}_{t}^{rs}|\mathbf{O}^{rs})] \quad (4)$$

$$\text{P-RNPL} = \frac{1}{R}\sum_{r=1}^{R}\text{NPL}^r \quad (5)$$

where $\mathbb{1}$ is an indicator function, r, s, t indicates random restarts, students, and practice opportunities, respectively. T_s is the total number of practices of student s, and D is the total number of practices of all students of current skill.

3.3 Consistency

Depending on different initial values of parameters, Knowledge Tracing and FAST can converge to points with different properties (such as plausibility or prediction of mastery). We favor a consistent model that has a low variance in properties across random restarts. Here, we extend the problem of Identifiability, where only global optimal log likelihood points are involved, into a more general problem of consistency, where all converged points are examined. The measurement of all converged points might be more operational in practice since it can be hard to judge whether the algorithm reaches a local or global optimum. For example, it is not clear how many random restarts are needed. Also, it is not sure whether converged points with log likelihood very close to the identified highest one can be treated as global optima or not.

Consistency of AUC, GS, NPL (C-RAUC, C-RGS, C-RNPL). Based on the explained importance of the performance metric AUC and the plausibility metrics GS and NPL, we think that a good model should also present low variance in these metrics across random restarts. As a result, we define consistency metrics C-RAUC, C-RGS, C-RNPL correspondingly by computing the standard deviation[1] of each single model's metrics across multiple random restart runs (r) on the test set with some transformation to map them into [0, 1] interval. Here, C- stands for consistency and R stands for random restarts. For example, for computing C-RAUC, we use the following formula:

$$\text{C-RAUC} = 1 - \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\text{AUC}^r - \overline{\text{AUC}})^2} \quad (6)$$

Consistency of the Predicted Probability of Mastery (C-RPM). Student models are usually used to assess whether and when students reach mastery, based on which tutoring systems give adaptive instruction. A model lacking consistency in mastery prediction will lead to varying decision in instruction, which can have a significant impact on students. So we also construct a metric to quantify this consistency, inspired by previous studies [2, 15, 7]. We use the conventional definition of Mastery as the probability of Learned reaching 0.95 [4]. We compute $\text{P}(\text{L}_t{=}\text{Learned}|\mathbf{O}_{1:t})$, the posterior knowledge estimation of being in the Learned state at t^{th} practice updated by 1^{st} to t^{th} practice observations $\mathbf{O}_{1:t}$. We also refer to it as $\text{P}(\widetilde{L}_t|\mathbf{O})$ for simplicity. We then compute the probability of reaching Mastery as the percentage of students predicted to ever have $\text{P}(\widetilde{L}_t|\mathbf{O}) \geq 0.95$, which means achieving a 0.95 posterior knowledge estimation in a practice sequence for the current skill. We refer to this probability as P(Mastery) or PM (Equation 7). We then compute the standard deviation of P(Mastery) across different runs, transform it to map to [0, 1] interval, and refer to it as C-RPM where C- stands for consistency, R stands for random restarts (Equation 8):

$$\text{PM}^r = \frac{1}{S}\sum_{s=1}^{S}\mathbb{1}\{\text{P}(\widetilde{L}_t^{rs}|\mathbf{O}^{rs}) \geq 0.95, \exists t \in [1, T_s]\} \quad (7)$$

$$\text{C-RPM} = 1 - \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\text{PM}^r - \overline{\text{PM}})^2} \quad (8)$$

where r, s, t indicates random restarts, students, and practice opportunities respectively. T_s is the total number of practices of student s of current skill.

Cohesion of the parameter vector space (C-RPV). Fixed point analysis has been used to show that we need all four parameters to define the overall behavior of Knowledge Tracing [19] during the prediction phase, when knowledge estimation is updated by prior observations. We use this conclusion to construct another consistency metric. To capture all four parameters, we construct a Euclidian vector based on the four fitted parameters Init, Learn, Guess, and Slip for each single model. For FAST, we compute the four parameters with only the intercepts in the logistic regression components after fitting with features during training. We then compute the Euclidian distance of each vector to the mean of the parameter vectors (similar to the cluster cohesion measurement), and then perform a transformation to map this value to [0, 1] interval. We define it as C-RPV where C- stands for consistency, R stands for random restarts, and PV stands for parameter vector:

$$\text{C-RPV} = 1 - \frac{1}{2R}\sum_{r=1}^{R}||\mathbf{V}^r - \overline{\mathbf{V}}|| \quad (9)$$

where $\mathbf{V}^r$ is the parameter vector of the r^{th} random restart. $\mathbf{V}^r$=(Init^r, Learn^r, Guess^r, Slip^r). $\overline{\mathbf{V}}$ is the mean of the parameter vectors across the random restarts.

3.4 Metric Selection

Our proposed Polygon framework consists of three dimensions: prediction, plausibility, and consistency, and allows flexibly designed metrics for each dimension. The metrics we introduced before are the potential ones to be considered. We propose a principled way to select metrics to instantiate the framework: selected metrics should cover all three dimensions while having the smallest pairwise correlation. To achieve this, we examine the scatterplot and correlation of each pair of the metrics and conduct a significance test. Finally, we report our selected metrics in Section 4.3.1.

4. STUDIES AND RESULTS

4.1 Datasets and Features

We conducted experiments on datasets from different tutoring systems: Geometry Cognitive Tutor [12], OLI Engineering Statics [18], Java programming tutor [10], and the

[1]We use uncorrected sample standard deviations to map the metric to [0, 1]. With a large enough sample size (100 in our study), the bias of this estimator is small. For a smaller sample size, the corrected version might be considered.

Physics tutoring instance of the BBN learning platform [14]. Table 1 shows descriptive statistics (#observations indicates the smallest assessable practice units of students).

Geometry, Statics. We obtained these datasets from PSLC Datashop [13]. The Geometry dataset has data from the area unit of the Geometry course, which was conducted during the 1996-1997 school year. The Statics dataset has data from multiple schools during Fall 2011. We defined a problem (item) by concatenating the problem hierarchy, problem name, and step name. We defined a skill by concatenating the problem hierarchy and original skills, and treated the combination of skills as one unique skill if multiple skills are associated with a problem. For the Statics dataset, we randomly selected 20 skills (from the total of 156) to avoid bias towards this dataset when we aggregate across datasets. We further removed 3 skills where there are fewer than 10 observations in total, resulting in 17 skills. For FAST models, we constructed binary item indicator features for each problem with fitted coefficients represent item difficulties. Such models have been known for their high predictive performance [11, 8], and we plan to examine other dimensions as well.

Java. The Java dataset was collected from an online Java programming tutoring system [10] from Fall 2010 to Fall 2014. For each problem, students are asked to give the value of a variable or the printed output of a Java program after they have executed the code in their mind, and the system assesses correctness. The Java programs are instantiated randomly from templates on every attempt. Students can make multiple attempts until they think they have mastered the skill, or just give up. Problems are grouped by Java topics (each problem is mapped to a single topic), and we considered each topic as a skill. We consider each problem template as a single item. For FAST models, we also constructed binary item indicator features, adding to the exploration of the effect of item difficulties.

Physics. The Physics dataset was collected from the BBN Learning Platform [14], a domain-independent, problem-solving-based online learning platform. Students can solve problems without any help, or request a decomposition of the problem into steps. The steps lead students through a carefully crafted directed path to help solve the problem. We used logs collected from 40 users solving 10 problems from the Electric Circuits units. Each of these problems and steps are annotated with electric circuits skills (in total 10). In addition to capturing student actions at the items, the platform logs requests for help, feedback received, and problem navigation actions. We derived 105 numeric features from these logs, performed feature selection, and finally used the top ranked feature for FAST. This allows us to inspect the effect of help in the Knowledge Tracing framework.

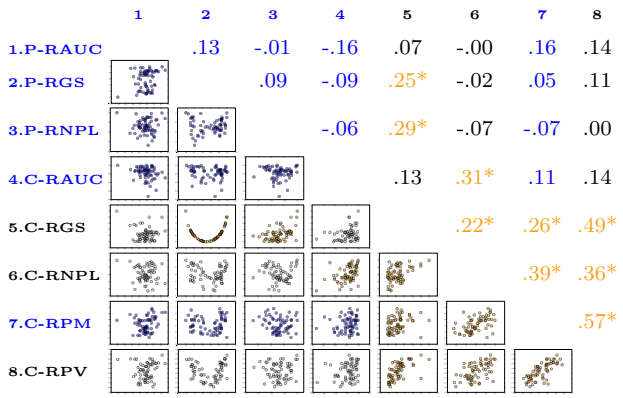
4.2 Experimental Setup

We used Expectation Maximization (EM) for training Knowledge Tracing, and Expectation Maximization with features for FAST [8]. We uniformly initialized each parameter within (0, 1) at each run for Knowledge Tracing, and we uniformly initialized each feature coefficient within (-10, 10) for FAST, which resulted in original parameters approximately covering (0, 1). We drew 100 different initial values for each parameter. We set 500 as the maximum EM iteration, 50 as the maximum LBFGS iteration and the log likelihood's relative change within 10^{-6} as convergence criteria. We trained each skill independently and used a user-stratified data split: 80% of the students were randomly selected into the training set, and the remaining students were assigned to the test set. In this way, models can be generalized to unseen students.

Table 1: Dataset descriptive statistics.

Dataset	#observations	#skills	#students	%correct
Geometry	5,055	18	59	75%
Statics	23,390	17	326	77%
Java	43,696	20	328	67%
Physics	10,063	10	40	62%

Table 2: Scatterplot and Kendall rank correlation among metrics of all skills (65) from Knowledge Tracing. Metrics selected into Polygon are shown in blue. Values shown in blue indicate a low correlation, and values shown in YellowOrange with asterisks indicate statistical significance (α=0.05).



4.3 Results

4.3.1 Metric Selection

In order to obtain a compact instantiation of the Polygon evaluation framework, we analyze the pairwise correlation among the proposed metrics on Knowledge Tracing models. For each skill we compute eight metrics based on 100 random restarts and analyze the relationship across skills. Table 2 shows that C-RGS, C-RNPL and C-RPV all include significant correlations with other metrics. Particularly, the scatterplot of P-RGS and C-RGS shows a U-shape; we think this finding is because the mean and standard deviation of Bernoulli-distributed variables (GS) have this property. Finally, we instantiate the **Polygon** framework with five metrics in our study: **P-RAUC**, **P-RGS**, **P-RNPL**, **C-RAUC** and **C-RPM**, where they cover three dimensions and have low, non-significant pairwise correlations.

4.3.2 Evaluation on Multiple Random Restarts

We now present how we use Polygon to evaluate multiple random restart models and single models on Knowledge Tracing and FAST. Figure 1 shows Polygon evaluation per dataset aggregated across skills. Overall, FAST mostly have Polygon areas covering that of Knowledge Tracing. Considering the variance across skills, FAST has significantly higher values in all five metrics (α=0.05, $p < 0.0001$ by Wilcoxon signed-rank test), suggesting that it might promise not only higher predictive performance, but also higher plausibility and consistency. One possibility is that the constructed features indirectly constrain the optimization algorithm to

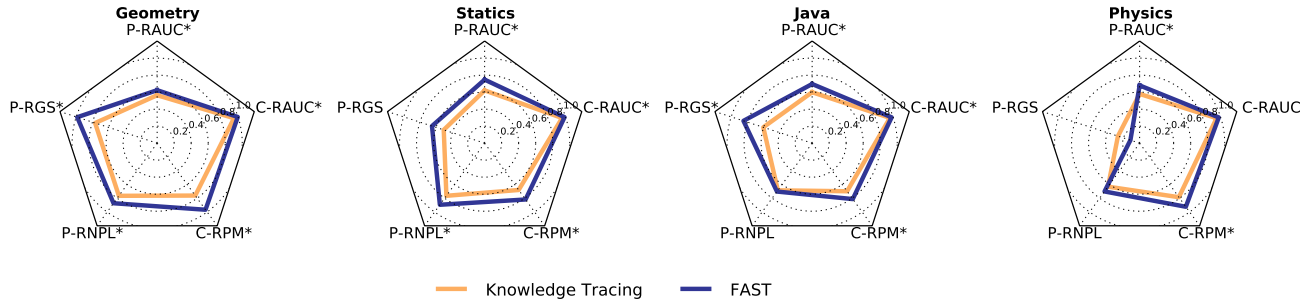


Figure 1: Polygon metrics per dataset comparing Knowledge Tracing and FAST. An asterisk (∗) indicates statistical significance under Wilcoxon signed-rank test (α=0.05). FAST's Polygon area mostly covers that of Knowledge Tracing.

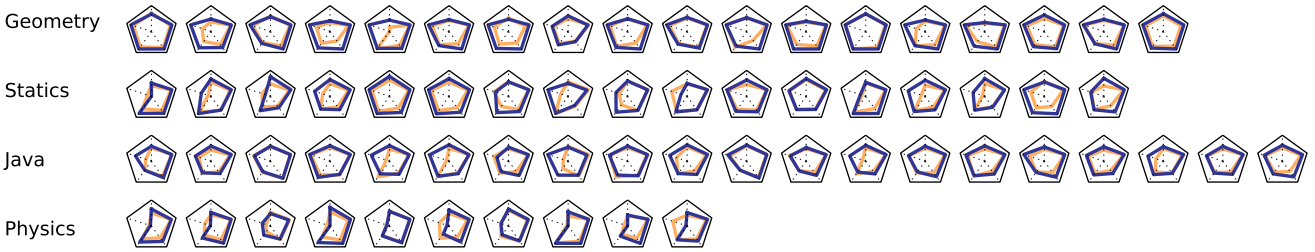


Figure 2: Polygon metrics per skill comparing Knowledge Tracing and FAST. FAST's Polygon area mostly covers that of Knowledge Tracing.

search within regions with both high fitness and plausibility. However, FAST's plausibility seems to be less stable, as compared to other properties, since its improvement varies across datasets.

We further examine Geometry, Statics and Java datasets where we use FAST with item difficulty features. As shown in Figure 1, FAST significantly outperforms Knowledge Tracing in all metrics, except for P-RGS on Statics and P-RNPL on Java, where FAST still presents positive tendencies. Generally speaking, using item difficulty features in Knowledge Tracing not only increases the model's predictive performance, but also its plausibility and consistency. However, the relative improvement in plausibility varies across datasets.

In the Physics dataset, FAST using problem decomposition requested features has a higher P-RAUC (significant), P-RNPL, C-RPM (significant), and C-RAUC, yet it also has a lower P-RGS, compared with Knowledge Tracing (not significant). Noticing that both methods have very low P-RGS, we suspect that skill definitions may be too coarse-grained, meaning that latter practices may involve potential new skills, where students fail more often than in the beginning. Thus, student models fitted from such data might be prone to estimating high Guess and Slip. FAST may be more vulnerable to bad skill definitions, since it might seek to fit the data as the primary goal, given that it has significantly higher predictive performance. In order to find out more about these potentially ill-defined skills, we further examine Polygon for each skill, as shown in Figure 2. This analysis shows that more than half of the skills in the Physics dataset have very low P-RGS, and particularly, there are two skills where FAST and Knowledge Tracing have an obvious gap on P-RGS (6^{th} and the last one), which should cause Knowledge Tracing to obtain a higher average value over FAST. We plan to examine whether refinement of the skill definitions will increase plausibility of both methods and FAST's relative quality for P-RGS in next steps.

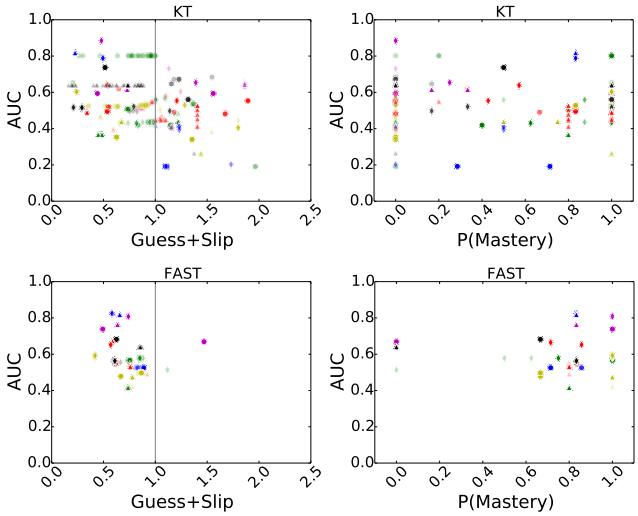


Figure 3: Evaluation on each skill's each random restart on Geometry dataset. Each color-shape corresponds to one skill. Each point corresponds to one random restart convergence point. Comparing with Knowledge Tracing, FAST generates more consistent, plausible models.

4.3.3 *Drill-down Evaluation of Single Models*

Polygon not only evaluates a method from multiple random restarts, but also contains components that can evaluate a single model. We use AUC, GS (Guess+Slip<1), and NPL to analyze each single model's predictive performance and plausibility, and also use the component PM (P(Mastery)) to get an intuitional understanding of a single model's effect on tutoring. Figure 3 visualizes AUC, Guess+Slip, and P(Mastery) of each random restart of each skill for Knowledge Tracing and FAST on Geometry dataset. Each color-shape corresponds to one skill, while each point corresponds to one random restart convergence point. We can easily determine different behaviors between Knowledge Tracing and FAST. FAST generates more consistent solutions than

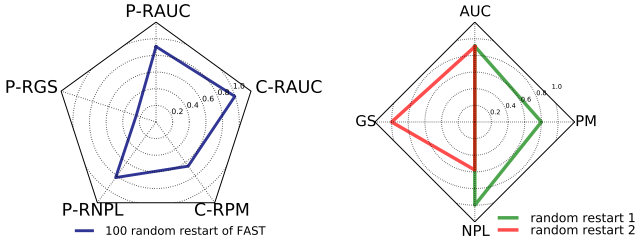


Figure 4: Polygon evaluation on a skill (id=154) on Statics dataset. The multi-model pentagon reveals this skill has high AUC consistency but low P(Mastey) consistency. The single-model quadrangle further reveals the contradictory properties of two random restart single models even they have very similar AUC.

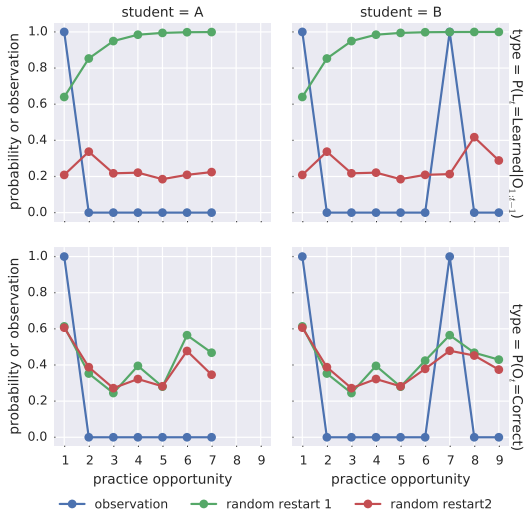


Figure 5: Comparison of two random restart single FAST models of a skill (id=154) from Statics dataset on two students. Both models have similar curves of predicted P(O_t=Correct) but have substantially different curves of predicted P(L_t=Learned | $\mathbf{O}_{1:t-1}$).

Knowledge Tracing, since there is less spread both horizontally and vertically of the random restart points within the same skill for all three metrics. FAST also generates more plausible models than Knowledge Tracing, since most of the points fall into Guess+Slip<1 region. Note that FAST asserts that students are more likely to reach mastery, since the converged points mostly lie in the higher-value region.

However, does FAST perform well on every skill? If not, can we use Polygon to effectively identify such skills and better understand the behavior? Based on previous skill-specific polygon evaluations (Figure 2), we identify one skill (3^{rd} polygon on the 2^{nd} row) on the Statics dataset, where Knowledge Tracing has better P-RGS than FAST. In Figure 4 the left-hand figure shows that this skill has a very high consistency of predictive performance (C-RAUC), yet a very low consistency of PM (C-RPM) across 100 random restarts. We further pick two of the random restarts and compute the polygon metrics for single models, as shown in Figure 4 right-hand single-model quadrangle. The quadrangle reveals that these two random restarts have almost identical AUC, yet have contradictory assertions about learning and mastery. In order to better understand the behavior, we

Table 3: Kendall rank correlation among single model AUC, GS, NPL and log likelihood (LL) on training set for the same skill across 100 random restarts on Knowledge Tracing. We report the number of skills and in the bracket the average of the correlation values across skills under each positive (+) or negative (-) correlation relation (zero correlation ignored) among all skills (65).

	AUC		GS		NPL	
	+	−	+	−	+	−
AUC			41(0.6)	23(-0.6)	35(0.7)	30(-0.5)
LL	46(0.5)	19(-0.4)	34(0.5)	30(-0.5)	30(0.4)	35(-0.5)

pick two students from each one of these random restarts, and plot the predicted correctness curve and knowledge level curve (conditioned on prior observations). Figure 5 shows a severe problem in comparing these two random restarts: they have very similar predicted correctness, yet present fundamentally different predicted knowledge levels. We think that this problem extends the identifiability problem, in the sense that similar predicted correctness curves though not identical can be problematic if the predicted knowledge level curves differ greatly. Also, we observe the empirical degeneracy of random restart 1: with more incorrect practices, the predicted probability of Learned increases. This analysis showcases the deficiency of using only predictive performance to evaluate student models, and the effectiveness of Polygon metrics in identifying hidden problems.

4.3.4 *Implications for Single Model Selection*

We further examine the deficiency of using prediction performance or fitness metrics to select single models. We compute the Kendall rank correlation between AUC and the plausibility metrics for each run of each skill of Knowledge Tracing. Table 3 shows the deficiency of using only AUC to select the best random restart. There are more than one-third of skills that show a negative correlation between predictive performance and plausibility across different runs, and the magnitude of the negative correlation on average is not small. What about choosing the model with the maximum likelihood (LL) on the training set? Table 3 also shows the correlation between LL, AUC, and the plausibility metrics across different random restarts. Overall, about 71% (46/65) of the time, choosing the maximum LL on the training set can lead to a higher predictive performance in the test set, yet we have no more than 46% (30/65) of the time to get a more plausible model. These findings show that LL fails to offer a better choice than AUC. We think that a practical generalizable way to obtain a latent variable student model with both high predictive performance and plausibility remains to be explored, and Polygon provides important insights.

5. CONCLUSIONS

In this paper, we propose a general unified evaluation framework (that we call Polygon) to evaluate student models with latent knowledge estimates. Prior studies have presented different properties of the estimated parameters of Knowledge Tracing, yet there are no unified, quantitative evaluations for general student models. Our primary contribution lies in the quantitative unification of three aspects for general latent variable student models: predictive performance, plausibility, and consistency. We propose novel metrics and present a principled way to select proper metrics. Our defined dimensions extend the definitions of previously defined Identifia-

bility and Model Degeneracy, which allows us to understand such problems more practically and more generally. A secondary contribution is that we show that a recent model with proper features, known as FAST, generally provides higher predictive models with higher plausibility and consistency than Knowledge Tracing. This suggests that proper features might help the optimization algorithm to constrain the search towards more plausible, more predictive regions.

There are several areas in which we can further extend our study. First, a single metric or perspective considering the multiple facets introduced in our analysis can further improve the accessibility of the evaluation. Also, each single metric can be further improved. For example, we can investigate the proper number of random restarts. However, Polygon's current individual metrics already provide insights for training student models. For example, incorporating the plausibility metric as a penalty into the optimization objective function can guide the algorithm to search within the high plausibility region. Second, external measurements applied in prior studies [4, 2, 6] may help to validate our framework. However, Polygon primarily serves as domain-independent internal validity, which is useful when external resources are not available. Third, the plausibility measurement can be a mixture of both student model and skill model evaluations. Will each model's relative quality be different when we examine well-defined vs. ill-defined skills? Can we utilize plausibility metrics to inspect skill model qualities? These are questions that remain unanswered. Fourth, we need to further understand and improve FAST. Since there are still cases where FAST generates models with low plausibility or low consistency, is there a principled way to construct features that maximize all three dimensions? Also, as we have only studied cases where a single feature (besides the intercept) is activated for each observation, will increasing the number of features change FAST's behavior?

Our study is still exploratory and serves as a first step towards a more theoretical, deeper understanding of the parameter estimation of complex latent variable student models. We hope that our work can open the door to more studies in the community on building student models that can yield not only better predictions of student performance but also more reliable, effective tutoring systems.

6. ACKNOWLEDGMENTS

This research is supported the Advanced Distributed Learning Initiative[2], Pearson[3] and the US Office of Naval Research (ONR) contract N00014-12-C-0535.

7. REFERENCES

[1] R. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems 2008*, pages 406–415. Springer.

[2] J. E. Beck and K.-m. Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146.

[3] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

[4] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[5] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *6th International Conference on Educational Data Mining*, pages 28–35, 2013.

[6] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.

[7] J. P. González-Brenes and Y. Huang. Your model is predictive— but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *Proceedings of the 8th Intl. Conf. on Educational Data Mining*, 2015.

[8] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proceedings of the 7th Intl. Conf. on Educational Data Mining*, 2014.

[9] G. Gweon, H.-S. Lee, C. Dorsey, R. Tinker, W. Finzer, and D. Damelin. Tracking student progress in a game-like learning environment with a monte carlo bayesian knowledge tracing model. In *Learning Analytics and Knowledge Conference 2015*.

[10] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning, 2010*.

[11] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, 2014.

[12] K. R. Koedinger. Geometry area (1996-97), February 2014. In *URL https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76*.

[13] K. R. Koedinger, R. S. J. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*, pages 43–55, Boca Raton, FL, 2010. CRC Press.

[14] R. Kumar, G. Chung, A. Madni, and B. Roberts. First evaluation of the physics instantiation of a problem-solving based online learning platform. In *Intl. Conf. on Artificial Intelligence in Education 2015*.

[15] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th Intl. Conf. on Educational Data Mining*, pages 118–125, 2012.

[16] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. *EDM*, 2010:161–170, 2010.

[17] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education*, pages 531–538.

[18] P. Steif and N. Bier. Oli engineering statics - fall 2011, February 2014. . In *URL https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507*.

[19] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.

[2] `http://www.adlnet.gov/`

[3] `http://researchnetwork.pearson.com/`

Predicting Student Performance In a Collaborative Learning Environment

Jennifer K. Olsen
Human Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
jkolsen@cs.cmu.edu

Vincent Aleven
Human Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
aleven@cs.cmu.edu

Nikol Rummel
Institute of Educational Research
Ruhr-Universität Bochum
Bochum, Germany
nikol.rummel@rub.de

ABSTRACT

Student models for adaptive systems may not model collaborative learning optimally. Past research has either focused on modeling individual learning or for collaboration, has focused on group dynamics or group processes without predicting learning. In the current paper, we adjust the Additive Factors Model (AFM), a standard logistic regression model for modeling individual learning, often used in conjunction with knowledge component models and tutor log data. The extended model predicts performance of students solving problems collaboratively with an ITS. Specifically, we address the open questions: Does adding collaborative features to a standard AFM provide a better fit than the standard AFM? Also, does the impact of these features change based on the nature of the knowledge (conceptual v. procedural) that is being acquired? In our extended AFM models, we include a variable indicating if students are working individually or in pairs. Also, for students working collaboratively, we model both the influence on learning of being helped by a partner and helping a partner. For each model, we analyzed conceptual and procedural datasets separately. We found that both collaborative features (being helped and helping) improve the model fit. In addition, the impact of these features differs between the collaborative and procedural datasets, suggesting collaboration may affect procedural and collaborative learning differently. By adding collaborative learning features into an existing regression model for individual learning over a series of skill opportunities, we gain a better understanding of the impact that working with a partner has on student learning, when working with a step-based collaborative ITS. This work also provides an improved model to better predict when students have reached mastery while collaborating.

Keywords

knowledge tracing, collaborative learning, educational data mining, Additive Factors Model

1. INTRODUCTION

The modeling of student knowledge has been shown to be an important aspect of Intelligent Tutoring Systems (ITSs) technology. A variety of modeling approaches have been used to model student knowledge and have often been used to support individualized learning [2, 3, 15, 25]. Models can provide an accurate prediction of learning and also provide insights into how people learn. However, these types of models typically account for students who work individually with an ITS; they do not account for situations in which students learn collaboratively in dyads or small groups, supported by ITS technology. Yet collaboration cannot be ignored since it has been shown to be beneficial for student learning [6, 19] and there may be relative strengths for collaborative and individual learning [11]. Students who work collaboratively may have different learning rates than when working individually; this effect may be caused from being helped by a partner or helping a partner. A key question is, therefore: How can modeling techniques used for individual learning be adapted so they help provide predictions and insights into collaborative learning, in addition to individual learning? Specifically, how can these models be adapted to account for the fact that the collaborating partners may influence each other's learning? What insight can models provide regarding this influence? In our ITS, students work either collaboratively or individually on the problem sets. We extend the Additive Factors Model (AFM) [2, 15] by including features that are unique to collaboration, in an attempt to better model both individual and collaborative learning.

Much of the research on learning prediction has focused on modeling individual learning such as through Bayesian Knowledge Tracing [3], AFM [2, 15], and Knowledge Decomposition Model [25]. These models accurately predict student performance and can advance our understanding of how students learn. Previous research has adapted these types of models to better predict and understand individual learning, such as by treating correct and incorrect attempts differently [15] or by including the transfer that may happen between similar skills [25]. For our work, we are using a version of the AFM. The AFM has frequently been used to assess and predict individual student performance. The AFM is a generalized logistic mixed model [1]. It is widely used to fit learning curves and to analyze and improve student learning [1]. To adapt the AFM to account for aspects of collaborative learning, we can apply the same types of principles that have been applied to increase our understanding of individual learning and apply them to collaborative learning. For example, individual models can account for the transfer of learning from previous similar opportunities [25]; the same method can be applied to collaborating students having an opportunity to learn from watching their partner solve steps.

Prior research within collaborative learning has focused on analyzing collaborative processes to better understand learning and social influence [5, 20]. Within this area, there are multiple approaches for better understanding the collaborative processes.

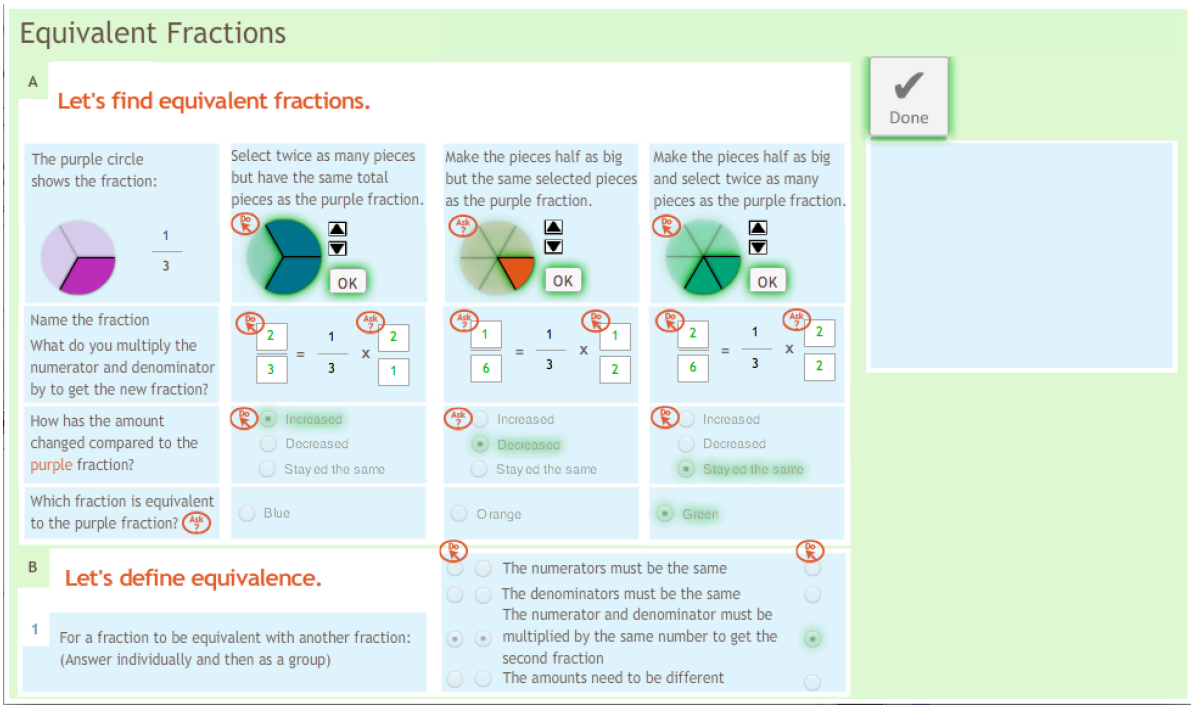


Figure 1. An example of a conceptual problem showing the different steps assigned to the partners in the collaborative condition based on the "Do" and "Ask" icons.

Some research aims to detect and classify collaboration skills, such as social deliberation skills and collaborative networks [21, 24]. Other research looks at the change in communication and processes that happen over time [10, 18]. Research has also focused on group dynamics and how we can recognize and intervene with groups that are not collaborating well [8, 9, 16]. Another aspect of collaboration that has been studied is asynchronous work that occurs on discussion boards and how this can influence learning and retention [22, 23]. Although this research is broad in the types of research questions that are addressed and covers many aspects of collaboration, much of the work does not attempt to predict student performance as students collaboratively solve problems. Such predictions could support student learning, for example by informing problem choices for dyads to help students where they are struggling. There has been previous work that has studied predicting performance by predicting posttest scores based on pair actions and found student interactions are predictive of the posttest score [17]; however, this work focuses on environments where the actions of collaborating students within a dyad or group cannot be distinguished (i.e., it is not known who took the action). In collaborative environments, in which the actions of the students within a collaborating group can be distinguished (e.g., a collaborative ITS), including collaborative features in models that have typically been used to predict individual performance may support a better understanding of the collaborative learning process and the ability to predict performance when students are collaborating. Previous work has attempted to address this issue by predicting performance of students based on their speech with an intelligent agent and found semantic match scores as a key predictor of later test performance [12]. Our work adds to this body of literature by investigating the prediction based on student actions within a system and how students will later do on similar items. The analysis of the student actions may provide different insights into the collaborative processes.

Extending the AFM with collaborative features enables us to study how collaboration might influence learning. Prior research with collaborative learning has shown that within mathematics, collaborative learning may better help students acquire conceptual knowledge, whereas individual learning activities may be more conducive to learning procedural knowledge [11]. Since our data set, obtained with a fractions tutor that supports collaborative learning, described below, includes both conceptual and procedural activities [13], we can study whether and how collaboration affects learning differently for these types of activities. By separately fitting models capturing collaborative and individual learning to data from procedurally versus conceptually oriented problems, we may be able to add to the understanding of how the different aspects of collaborative learning may have different strengths for different types of knowledge.

In this paper, we extend the AFM to (a) distinguish the learning that may occur when working individually versus collaboratively and (b) to capture learning that may occur from observing a partner's answers to steps. We also explore (c) whether the effect of these features is different in activities designed to support learning of concepts, compared to activities designed to support learning of procedures. By modeling student knowledge when working collaboratively, we aim to develop a better understanding of collaborative and individual learning processes. An improved model would also allow us to more accurately predict student performance and has the potential to support learning more effectively within an ITS, for example through improved problem selection for collaborative learning.

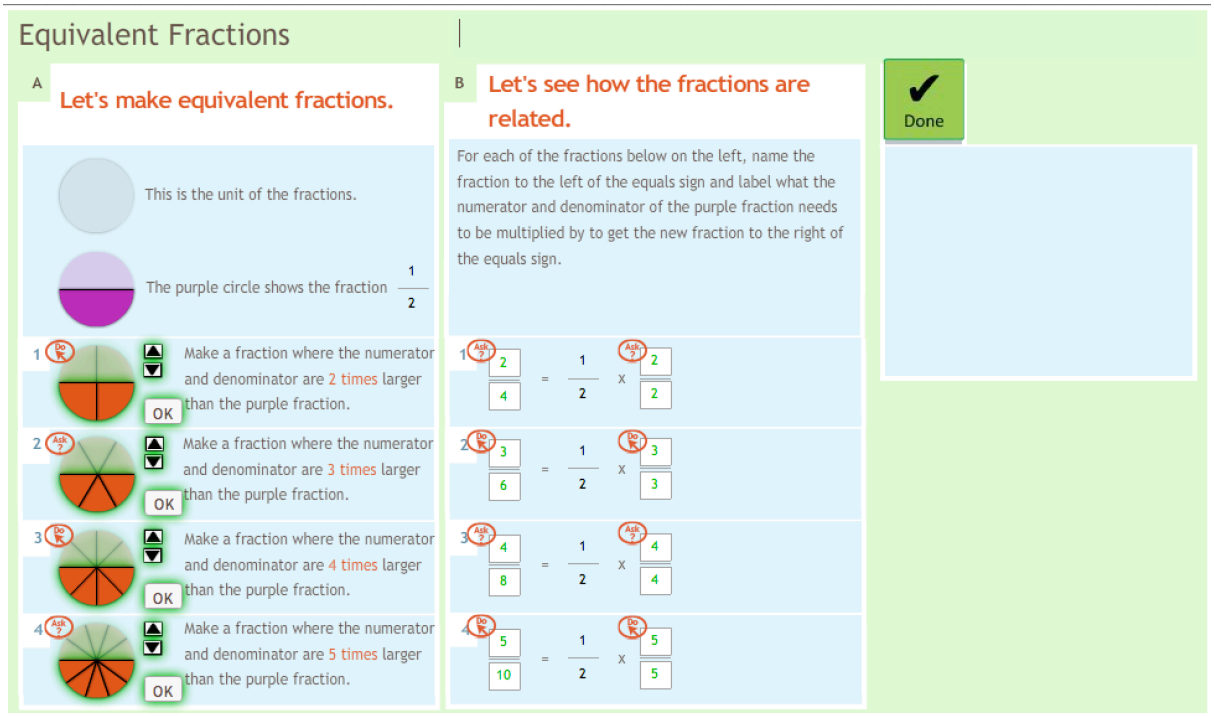


Figure 2. An example of a conceptual problem showing the different steps assigned to the partners in the collaborative condition based on the "Do" and "Ask" icons.

2. METHODS

In the following sections, we present the collaborative ITS for fractions learning that was used in our study and explain the experimental set-up that was used for data collection.

2.1 Individual and Collaborative Fraction Tutors

In the study that produced the data set that we analyze in the current work, students worked with an ITS that targeted equivalent fractions knowledge either working individually or with a partner. We developed two parallel versions of a fractions tutor, one with embedded collaboration scripts and one for individual learning. We created all tutor versions using the Cognitive Tutor Authoring Tools (CTAT), which we extended so it supports the authoring of tutors with embedded (static) collaboration scripts that are tied to the problem state [14]. Both the individual and the collaborative tutor versions had procedural and conceptual problem sets. Figure 1 shows an example of a conceptual problem, which shows the student different relationships between the numerators and denominators and that only the one where the amount stays the same shows an equivalent fraction. On the other had, Figure 2 shows an example of a procedural problem where the student makes equivalent fractions by multiplying the numerator and denominator by the same number. The individual ITS provides standard ITS support (step-level guidance for problem solving, with correctness feedback, next-step hints, and error-specific feedback messages) while the collaborative ITS also has embedded collaborative scripts. The students working collaboratively did so through a synchronous, networked collaboration. That is, collaborating students sat at their own computer and had a shared (though differentiated) view of the problem state. They could discuss the activity through audio by using Skype.

The collaboration was supported through proven collaboration scripts such as the use of roles, cognitive group awareness, and unique information, embedded in the interactions with the ITS. First, the embedded collaboration scripts defined roles that distribute the activities between the students and provide guidance to the students about what they should be doing to interact with their partner and help to scaffold this interaction. A second collaborative support feature we used in the collaborative problem sets is cognitive group awareness. Cognitive group awareness means that group members have information about other group members' knowledge, information, or opinions and has been shown to be effective for the collaboration process [7]. The last collaborative support feature is the use of unique information to create a sense of individual accountability. Individual accountability means that each group member takes responsibility for the group reaching its goal [19]. All of these collaboration features, as implemented, assigned different problem steps to each student within a collaborating dyad. The "Do" and "Ask" icons shown in Figures 1 and 2 indicate which student was responsible for solving a given step and which student had the role of supporting the other student; on the screen of the collaborating partner, the "Do" and "Ask" icons would be flipped. Therefore, problem steps divide into a student's own steps and that student's partner's steps. This distinction is important because, we will see, our extended AFMs treat these steps differently.

Our ITS is uncommon in that it was developed to support both collaborative and individual learning. This means that our data logs contain both records of individual and collaborative sessions, with a common set of features that is typical of ITS log data. (The data from the collaborative sessions were captured as separate streams from each student, where a partner's actions are not associated with a student's id.) Although the collaborative tutor had three different types of support for collaboration, each

scaffolding the interactions between the students in different ways, each of these support type led to the same pattern of information in the log data. For every step in a tutor problem, one student was responsible for answering the step and the other student's role was to monitor and help; therefore, the steps in the log data can be assigned to one partner or the other. Although not all collaboration environments allow for the distinction between student actions within a group, many environments can record this data and would then have similar log data to what we have, possibly even when student roles are not as clearly defined and supported.

2.2 Data Source

Our data is a set of collaborative and individual data that had been collected from a study [13] in which 4th and 5th grade students engaged in a problem-solving activity with the ITS for fractions learning described above. The experiment was a pull-out design, in which the students left their normal instruction during the school day to participate in the study. The data set comprises 84 students. Each teacher paired the students participating in the study based on students who would work well together and had similar math abilities. These pairs were then randomly assigned to one of four conditions: collaborative conceptual, collaborative procedural, individual conceptual, and individual procedural. Twice as many students were assigned to the collaborative conditions as to the individual conditions, so that the number of dyads in the collaborative conditions equaled the number of individual students in the individual conditions. Each student or dyad worked with the tutor for 45 minutes in a lab setting at their school during the school day.

We analyzed all tutor problems in terms of the underlying knowledge components (KCs) related to fraction equivalence. For the four conditions, the KCs were the same between the individual and collaborative conditions, but there was no overlap in the KCs between the conceptual and procedural items, as conceptual and procedural KCs were modeled separately.

3. MODELS

In this section we review the standard AFM and then present the models we made by adding collaborative features to this model.

3.1 Additive Factors Model

We first present the standard AFM, because this model is the basis on which all of our other models are built. The AFM [2] shows that the log-odds that a given student correctly solves a given step in a problem are a function of three parameters capturing, respectively, the given student's proficiency, the ease of the given knowledge component (KC, the skill the student is learning), and the learning rate. It assumes that the learning rate differs by KC but, for any given KC, is equal for all students. It further assumes that students differ in their general proficiency but in a way that affects all KCs and KC opportunities equally.

The AFM is a generalized mixed model. p_{ij} is the probability that student i gets step j right, θ_i is the random effect representing the proficiency of the student i. The fixed effect portion of the model includes β_k (the ease of KC k), γ_k (the learning rate of this KC), and N_{ik} (the prior learning opportunities the student had to apply KC k). The Q_{kj} term represents if an item the student encounters (i.e., a step in a tutor problem) uses KC k.

$$\ln\frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\gamma_k N_{ik}) \quad (1)$$

The standard AFM presented in Formula 1 is based on individual learning parameters of the opportunities that the individual has had with the KC. For the individual learning condition, these are all steps the student encountered in which the given KC applies. When this model is applied to the collaborative learning condition, on the other hand, these are the steps with the given KC that the given student is responsible for solving. This model however does not take into account that the learning rate for students may be different when working in a group compared to individually or that the students may learn from watching their partner solve problems.

3.2 Additive Factors Model with Condition

To investigate the difference in learning rates that may occur when students work individually, as compared to working in pairs, we added a feature to the original AFM that changes the slope based on condition (individual v. collaborative). Similar to the assumption that students learn at different rates from correct and incorrect answers in Pavlik, Cen, and Koedinger's Performance Factors Analysis, PFA [15], students may learn different amounts (per opportunity) when they are working individually versus collaboratively. In the collaborative condition, students are talking with their partner (through Skype) while solving steps that have been assigned to them. Having a partner may have an influence on their learning, even on steps that they (and not their partner) are responsible for solving. A student may get more learning out of a step they solved because of fruitful discussion with the partner, but could conceivably also learn less than when solving the step alone, with tutor help only, for example if the partner simply tells them the answer and the student does not reflect on the answer. In Formula 2, we capture the influence that the presence of a partner has on the student's own opportunities. A term c is added to represent the condition that the student is in at a given step. This allows the learning rate of a KC, γ_{kc}, to vary depending on the condition, so as to capture a difference in the learning that occurs between individual and collaborative work, on the student's own steps

$$\ln\frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\gamma_{kc} N_{ikc}) \quad (2)$$

By adding the condition parameter to the model, we can capture any differences in learning rates that may occur between working individually and within a group.

3.3 Additive Factors Model with Partner Opportunities

Within collaborative learning, there is an opportunity for students to learn from their partner's actions. Recall that when students work collaboratively in our tutoring system, the students are assigned to different roles for any given step (either solve it or help the partner solve it). Therefore, steps in tutor problems classify as the student's own steps or the partner's steps. On the partner's steps the student is watching and possibly providing advice, feedback, and explanations, which may create a learning opportunity for that student, even though he or she is not solving this step. Thus, we need to model the learning that occurs not only

Table 1. Prediction accuracy for the individual and collaborative procedural dataset across all models. The asterisks indicates the model with the best performance for that criterion.

Procedural Models	Log Likelihood	RMSE	AIC	Parameters
Standard AFM	-2010.34	0.4738	4080.69	30
AFM with Condition	-1983.39	0.4717	4056.77*	45
AFM with Partner Opportunities	-1984.59	0.4712	4059.17	45
AFM with Condition and Partner Opportunities	-1972.97*	0.4674*	4065.94	60

on a student's own opportunities (as modeled in Formulas 1 and 2) but also on their partner's opportunities. Learning on partner opportunities may be analogous to the learning decomposition that happens as students learn reading and their learning of a certain word benefits from seeing words with identical stems [25]. Although the student is not interacting directly with the tutor, there may still be learning. We assume that the learning that occurs when watching and/or helping a partner is possibly different from that which occurs when *doing* steps. We therefore added a new fixed parameter that takes into account the learning that could happen on a partner's opportunities. In the model seen in Formula 3, $\rho_k N_{lk}$ represents the learning (with its own learning rate) from a partner's opportunities on KC k (N_{lk}).

$$\ln\frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\gamma_k N_{ik}) + \sum_k Q_{kj}(\rho_k N_{lk}) \quad (3)$$

By adding the learning from partner's opportunities to the model, we can capture how students learn from their partner's opportunities, when their role is to observe and provide help and advice. This provides insights into the importance of helping a partner's work. The model also may provide better predictions of student performance when working in a collaborative condition where the student's actions can be differentiated.

3.4 Additive Factors Model with Condition and Partner Opportunities

The final model combines the collaborative features of the previous two. This model takes into account both the differences in learning rates that may occur for a student's own opportunities between individual and collaborative learning (captured in Formula 2) and also includes the learning that may occur by observing a partner's opportunities while working collaboratively (captured in Formula 3). Please note that the c (condition term) was not included in the partner's opportunities, because students who work individually do not have any partner opportunities to observe, making the partner opportunities always be 0 for students working individually.

$$\ln\frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\gamma_{kc} N_{ikc}) + \sum_k Q_{kj}(\rho_k N_{lk}) \quad (4)$$

This model combines the collaborative features of the previous two models to capture how these two ways of possibly benefitting from collaboration might balance.

4. RESULTS

For our analysis of the models, we evaluated the data from the procedurally-oriented tutor problems and the conceptually-oriented tutor problems separately to be able to see if the collaborative features that were added to the model have different effects for these two types of knowledge. Because students were assigned to either work on procedurally or conceptually oriented problems, there was no overlap in the students in the two datasets. Additionally, there was no overlap in the KCs in the datasets since any given KC captured either procedural or conceptual knowledge. With neither an overlap in students nor KCs between the datasets, the datasets can be analyzed separately, so as to analyze how collaboration (versus individual learning) might influence the learning of conceptual and procedural knowledge differently.

We measured the prediction accuracy of all of the models across the two data groups using the log likelihood, the root mean squared error on the training set (RMSE), and the Akaike information criterion (AIC). The log likelihood and RMSE provide a measure of fit not taking into account the complexity of the model. The AIC takes into account the complexity of the model when determining the fit of the model; it imposes a penalty based on the number of parameters. All of the models were run through a LIBLINEAR library in C [4]. Although in a standard AFM, the learning rate is restricted to be greater than or equal to zero, this restriction was not enforced in our models.

4.1 Procedurally-Oriented Problems

On the procedural dataset (see Table 1), the more complex models (i.e., the models that capture the influence of working with a partner in the ways discussed above) have a better fit in terms of log likelihood and RMSE, compared to the standard AFM. When comparing the models based on the AIC, all of the models that model aspects of collaborative learning have an improved AIC over the standard AFM. The AFM with Condition has the best AIC fit. Since the parameters are the same for the AFM with Condition and the AFM with Partner Opportunities, yet the former has a lower AIC, the condition the students are working in may be a better predictor of performance than having additional opportunities to observe a partner solving a step. Put differently, on procedural problems, having partner help when solving a step may influence learning more than helping a partner solve a step. It should be noted, however, that the difference in AIC between the two models is very small. The AIC for the model that combines the two collaborative features (AFM with Condition and

Table 2. Prediction accuracy for the individual and collaborative conceptual dataset across all models. The asterisks indicates the model with the best performance for that criterion.

Conceptual Models	Log Likelihood	RMSE	AIC	Parameters
Standard AFM	-1383.81	0.4815	2843.61	38
AFM with Condition	-1362.72	0.4804	2839.44	57
AFM with Partner Opportunities	-1359.67	0.4815	2833.33*	57
AFM with Condition and Partner Opportunities	-1344.50*	0.4772*	2841.01	76

Partner Opportunities) is higher, even though the log likelihood and RMSE are lower, indicating that the complexity of the model out-weighs the added gains.

4.2 Conceptually-Oriented Problems

For the models that were run on the conceptual dataset (see Table 2), the more complex models (i.e., those modeling how collaboration might influence learning) again have a better fit in terms of log likelihood and RMSE. As with the procedural dataset, these results indicate the importance of both the condition the students are working in (i.e., influence of partner help on the student's own opportunities) and of the partner opportunities (i.e., influence of helping a partner). When comparing the models based on the AIC, all of the models with collaborative features have an improved AIC over the standard AFM, and the AFM with Partner Opportunity has the best fit. Unlike with the procedural dataset, on conceptual problems, being able to observe a partner solving a step has more of an impact on predicted performance than condition.

5. DISCUSSION

AFMs are widely used models for predicting student performance. However, these models have mostly been used to predict the performance of students who are working individually. Students who are working collaboratively may go through different learning processes as they interact with other students, which currently are not accounted for in the standard AFM. In this paper, we wanted to see if adding collaborative features to AFM had an impact on the accuracy of the predicted learning performance of students in ITSs. Specifically, we investigated two mechanisms by which collaboration might influence learning. First, students might have different learning gains on steps they are responsible for solving because of the influence of a partner, such as through productive discussion or by being distracted. Second, a student might benefit from collaboration through engaging in discussion with a partner on steps that the partner is solving or by observing a partner as the partner solves the step. These two mechanisms were tested by two different ways of extending the AFM. First, we took into account the condition the student is working in (collaborative v. individual) by allowing the learning slope to vary based on condition. Second, we included the partner opportunities to capture the learning that may occur from observing/discussing a partner's answers to steps. These different learning mechanisms may differ for students who are working to acquire different types of knowledge. To take this into account, we analyzed our datasets for conceptual and procedural knowledge separately.

We first investigated if there is a difference between the learning rate of students working individually and those working collaboratively. To model the effect a partner may have on the steps that a student is responsible for solving, we added condition as a feature to the learning slope parameter. For both the procedural dataset and the conceptual dataset, the models that included condition outperformed the standard AFM based on AIC and log likelihood. Condition may be a useful predictor to include in a model for performance when students work collaboratively (or even, alternate between working collaboratively or individually) to more accurately predict performance.

To answer the question if observing and working with a partner on the partner's opportunities has an impact on learning (the second mechanism by which collaborative learning might help), we added an additional learning slope for a partner's opportunities to the standard AFM. Again, for both the procedural and conceptual datasets, the models that included the partner's opportunities outperformed the standard AFM based on AIC and log likelihood. This indicates that observing and helping a partner solve problems has an impact on a student's learning when working on either procedurally oriented problems or conceptually oriented problems. A partner's opportunity to practice a KC may be important to include in a learning model where students have the potential to work with another student.

Although the models built on the procedural and conceptual datasets cannot be compared directly, we can observe some differences in the order of the model fits that may indicate differences in the importance of different learning processes when acquiring different types of knowledge. The best model for the procedural dataset was the AFM with Condition, whereas the best fitting model for the conceptual dataset was the AFM with Partner Opportunity. These differences in the best-fitting model may indicate that collaboration might influence learning differently when learning procedural knowledge than when learning conceptual knowledge. When students are acquiring conceptual knowledge, observing a partner or helping a partner solve a step may have more of an impact than when a student is acquiring procedural knowledge.

The work makes a number of contributions to the field of EDM. It is one of the few to address how standard student modeling techniques in EDM can be applied to collaborative learning. Our modified AFM model predicts student performance as students collaboratively solve problems. The model can be applied to learning in collaborative environments in which the actions of the students within a collaborating group can be distinguished. The work extends the AFM so it can be applied to collaborative learning, capturing two different mechanisms by which collaboration might help students learn with a collaborative ITS. By applying these new models to a data set on both collaborative and individual learning, the work demonstrates that these two mechanisms might both be at work in conceptual and procedural learning, although to varying degrees. These findings contribute to enhance the understanding of the relative strengths of collaborative and individual learning.

A limitation of this dataset is that we do not have a comparison between the difficulty of the procedural and conceptual datasets. Any differences between the models for these datasets may not be due to the type of knowledge that is being acquired but may be related to where the students were in the learning process for these different types of data while learning. For future work, we are interested in using these models for student data where the students switch between working individually and collaboratively on the same sets of KCs, both conceptual and procedural. By modeling this data using the new AFMs we have created, we can better understand how the models will generalize to a more natural learning situation in the classroom. In addition, the models can be applied to situations where students come to the collaboration with different skills to see how students learn the skills from their partner. The AFMs with the added parameters provide improved models to better predict when students have reached mastery while collaborating or working individually.

6. ACKNOWLEDGMENTS

We thank the CTAT team, Daniel Belenky, Ryan Carlson, and Michael Yudelson for their help. This work was supported by Graduate Training Grant # R305B090023 and by Award # R305A120734 both from the US Department of Education (IES).

7. REFERENCES

[1] Boeck, P. 2008. Random Item IRT Models. *Psychometrika*, 73(4), 533–559.

[2] Cen, H., Koedinger, K. R., and Junker, B. 2007. Is Over Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. *In Proc. AIED*, pages 511–518.

[3] Corbett, A. T. and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278.

[4] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. 2008. LIBLINEAR: A library for large linear classification, *Journal of Machine Learning Research* 9(2008), 1871-1874.

[5] Janssen, J., & Bodemer, D. 2013. Coordinated computer-supported collaborative learning: Awareness and awareness tools. *Educational Psychologist*, 48(1), 40-55.

[6] Li, Y., Wang, J., Liao, J., Zhao, D., and Huang, R. 2007. Assessing collaborative process in CSCL with an intelligent content analysis toolkit. In *ICALT*.

[7] Lou, Y., Abrami, P. C., & d'Apollonia, S. 2001. Small group and individual learning with technology: A meta-analysis. *Review of educational research*,*71*(3), 449-521.

[8] Martinez-Maldonado, R., Yacef, K., & Kay, J. 2013. Data Mining in the Classroom: Discovering Groups' Strategies at a Multi-tabletop Environment. *Proc. EDM*, *2013*.

[9] McNely, B. J., Gestwicki, P., Hill, J. H., Parli-Horne, P., & Johnson, E. 2012. Learning analytics for collaborative writing: a prototype and case study. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 222-225). ACM.

[10] Mercer, N. 2008. The seeds of time: Why classroom dialogue needs a temporal analysis. *The Journal of the Learning Sciences, 17*(1), 33-59.

[11] Mullins, D., Rummel, N., & Spada, H. 2011. Are two heads always better than one? Differential effects of collaboration on students' computer-supported learning in mathematics. *International Journal of Computer-Supported Collaborative Learning*, *6*(3), 421-443.

[12] Nye, B. D., Hajeer, M., Forsyth, C. M., Samei, B., Millis, K., & Hu, X. 2014. Exploring real-time student models based on natural-language tutoring sessions.

[13] Olsen, J. K., Belenky, D. M., Aleven, V., & Rummel, N. 2014. *Using an intelligent tutoring system to support collaborative as well as individual learning. In 12th International Conference on Intelligent Tutoring Systems*, 134-143. Springer International Publishing.

[14] Olsen, J. K., Belenky, D. M., Aleven, A., Rummel, N., Sewall, J., & Ringenberg, M. 2014. Authoring Tools for Collaborative Intelligent Tutoring System Environments. *In 12th Int'l Conference on Intelligent Tutoring Systems.*

[15] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. 2009. Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.

[16] Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. 2009. Clustering and sequential pattern mining of online collaborative learning data. *Knowledge and Data Engineering, IEEE Transactions on*, *21*(6), 759-772.

[17] Rafferty, A. N., Davenport, J., & Brunskill, E. 2013. Estimating Student Knowledge from Paired Interaction Data. *Proc. EDM*.

[18] Reimann, P.: Time is precious 2009 Variable-and event-centred approaches to process analysis in CSCL research. International Journal of Computer-Supported Collaborative Learning, 4(3), 239-257.

[19] Slavin, R. E. 1996. Research on cooperative learning and achievement: What we know, what we need to know. *Contemporary educational psychology*, 21(1), 43-69.

[20] Stahl, G., Koschmann, T., and Suthers, D.. 2006. Computer supported collaborative learning: An historical perspective. *In R. K. Sawyer, editor, Cambridge Handbook of the Learning Sciences*. Cambridge University Press.

[21] Suthers, D., & Chu, K. H. 2012. Multi-mediated community structure in a socio-technical network. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 43-53). ACM.

[22] Wen, M., Yang, D., & Rosé, C. P. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us. *Proceedings of Educational Data Mining*.

[23] Wise, A. F., & Chiu, M. M. 2011. Analyzing temporal patterns of knowledge construction in a role-based online discussion. International Journal of Computer-Supported Collaborative Learning, 6(3), 445-470.

[24] Xu, X., Murray, T., Woolf, B. P., & Smith, D. 2013. Mining Social Deliberation in Online Communication--If You Were Me and I Were You. In *Educational Data Mining*.

[25] Zhang, X., Mostow, J., & Beck, J. E. 2007. All in the (word) family: Using learning decomposition to estimate transfer between skills in a Reading Tutor that listens. In *AIED2007 Educational Data Mining Workshop* (pp. 80-87).

Learning Instructor Intervention from MOOC Forums: Early Results and Issues

Muthu Kumar Chandrasekaran[1]
Min-Yen Kan[1]
Bernard C.Y. Tan[2]
Kiruthika Ragupathi[3*]

[1] Web IR / NLP Group (WING)
[2] Department of Information Systems
[3] Centre for Development of Teaching and Learning
National University of Singapore
{muthu.chandra, kanmy}@comp.nus.edu.sg, {pvotcy, kiruthika}@nus.edu.sg

ABSTRACT

With large student enrollment, MOOC instructors face the unique challenge in deciding when to intervene in forum discussions with their limited bandwidth. We study this problem of *instructor intervention*. Using a large sample of forum data culled from 61 courses, we design a binary classifier to predict whether an instructor should intervene in a discussion thread or not. By incorporating novel information about a forum's type into the classification process, we improve significantly over the previous state-of-the-art.

We show how difficult this decision problem is in the real world by validating against indicative human judgment, and empirically show the problem's sensitivity to instructors' intervention preferences. We conclude this paper with our take on the future research issues in intervention.

Keywords

MOOC; Massive Open Online Course; Instructor Intervention; Discussion Forum; Thread Recommendation

Categories and Subject Descriptors

H.3.3. [**Information Search and Retrieval**]: Information filtering; K.3.1. [**Computers and Education**]: Computer Uses in Education

1. INTRODUCTION

MOOCs scale up their class size by eliminating synchronous teaching and the need for students and instructors to be co-located. Yet, the very characteristics that enable scalability of massive open online courses (MOOCs) also bring significant challenge to its teaching, development and management [7]. In particular, scaling makes it difficult for instructors to interact with the many students — the lack of interaction and feelings of isolation have been attributed as reasons for why enrolled students drop from MOOCs [9].

MOOC discussion forums are the most prominent, visible artifact that students use to achieve this interactivity. Due to scale of contributions, these forums teem with requests, clarifications and social chatting that can be overwhelming to both instructors and students alike. In particular, we focus on how to best utilize instructor bandwidth: with a limited amount of time, which threads in a course's discussion forum merit instructor intervention? When utilized effectively, such intervention can clarify lecture and assignment content for a maximal number of students, promoting the enhancing the learning outcomes for course students.

To this end, we build upon previous work and train a binary classifier to predict whether a forum discussion thread merits instructor intervention or not. A key contribution of our work is to demonstrate that prior knowledge about forum type enhances this prediction task. Knowledge of the enclosing forum type (i.e., discussion on *lecture*, *examination*, *homework*, etc.) improves performance by 2.43%; and when coupled with other known features disclosed in prior work, results in an overall, statistically significant 9.21% prediction improvement. Additionally, we show that it is difficult for humans to predict the actual interventions (the gold standard) through an indicative manual annotation study.

We believe that optimizing instructor intervention is an important issue to tackle in scaling up MOOCs. A second contribution of our work is to describe several issues pertinent for furthering research on this topic that emerge from a detailed analysis of our results. In particular, we describe how our work at scale details how personalized and individualized instructor intervention is — and how a framework for research on this topic may address this complicating factor through the consideration of normalization, instructor roles, and temporal analysis.

2. RELATED WORK

While the question of necessity of instructor's intervention in online learning and MOOCs is being investigated [12, 20], technologies to enable timely and appropriate intervention are also required.

*This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

The pedagogy community has recognized the importance of instructor intervention in online learning prior to the MOOC era (e.g., [10]). Taking into consideration the pedagogical rationale for effective intervention, they also proposed strategic instructor postings: to guide discussions, to wrap-up the discussion by responding to unanswered questions, with "Socrates-style" follow-up questions to stimulate further discussions, or with a mixture of questions and answers [13]. However, these strategies must be revisited when being applied to the scale of typical MOOC class sizes.

Among works on forum information retrieval, we focus on those that focus on forum moderation as their purpose is similar to the instructor's role in a course forum. While early work focused on automated spam filtering, recent works shifted focus towards curating large volumes of posts on social media platforms [4] to distil the relevant few. Specifically, those that strive to identify thread solvedness [21, 8] and completeness [3] are similar to our problem.

Yet all these work on general forums (e.g., troubleshooting, or threaded social media posts) are different from MOOC forums. This is due to important differences in the objectives of MOOC forums. A typical thread on a troubleshooting forum such as Stack Overflow is centers on questions and answers to a particular problem reported by a user; likewise, a social media thread disseminates information mainly to attract attention. In contrast, MOOC forums are primarily oriented towards learning, and also aim to foster learning communities among students who may or may not be connected offline.

Further, strategies for thread recommendation for students such as [23] may not apply in recommending for instructors. This difference is partially due to scale: while the number of students and threads are large, there are few instructors per course. In this case, reliance on collaborative filtering using a user–item matrix is not effective. Learning from previous human moderation decisions [2], therefore, becomes the most feasible approach. Prior work on MOOC forums propose categorisation of posts [16, 5, 19] to help instructors identify threads to intervene. Chaturvedi *et al.* [5], the closest related work to ours, show each of the four states of their sequence models to predict instructor intervention to be distributed over four post categories they infer. In this paper, we use their results for comparison.

Different from previous works, we propose thread–level categories rather than post–level categories, since an instructor needs to first decide on a thread of interest. Then they need to read its content, at least in part, before deciding whether to intervene or not. We make the key observation that show thread–level categories identified as by the forum type, help to predict intervention.

Previous work has evaluated only with a limited number of MOOC instances. One important open question is whether those reported results represent the diverse population of MOOCs being taught. In this paper, we address this by testing on a large and diverse cross-section of Coursera MOOC instances.

3. METHODS

We seek to train a binary classifier to predict whether a MOOC forum thread requires instructor intervention. Given a dataset where instructor participation is labeled, we wish to learn a model of thread intervention based on qualities (i.e., features) drawn from the dataset. We describe our dataset, the features distilled used for our classifier, how we obtain class labels, and our procedure for instance weighting in the following.

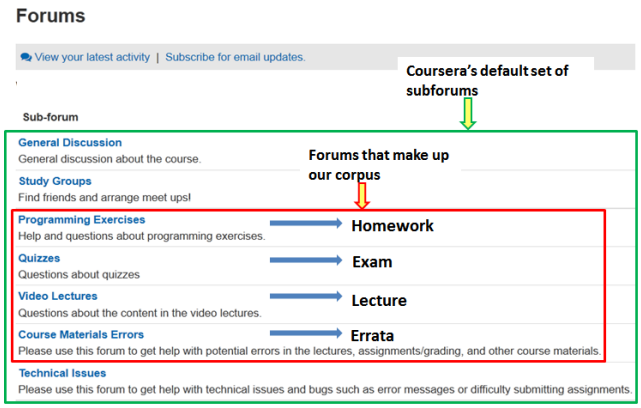


Figure 1: Typical top-level forum structure of a Coursera MOOC, with several forums. The number of forums and their labels can vary per course.

Forum type	All		Intervened	
	# threads	# posts	# threads	# posts
D61 Corpus				
Homework	14,875	127,827	3993	18,637
Lecture	9,135	64,906	2,688	10,051
Errata	1,811	6,817	654	1,370
Exam	822	6,285	405	1,721
Total	**26,643**	**205,835**	**7,740**	**31,779**
D14 Corpus				
Homework	3,868	31,255	1,385	6,120
Lecture	2,392	13,185	1,008	3,514
Errata	326	1,045	134	206
Exam	822	6,285	405	1,721
Total	**7,408**	**51,770**	**2,932**	**11,561**

Table 1: Thread statistics from our 61 MOOC Coursera dataset and the subset of 14 MOOCs, used in the majority of our experiments.

3.1 Dataset

For our work, we collected a large-scale, multi-purpose dataset of discussion forums from MOOCs. An important desideratum was to collect a wide variety of different types of courses, spanning the full breadth of disciplines: sciences, humanities and engineering. We collected the forum threads [1] from 61 completed courses from the Coursera platform[2], from April to August 2014, amounting to roughly 8% of the full complement of courses that Coursera offers[3].

For each course, we first assigned each forum[4] to one of several types based on the forum's title. For this study we focus on threads that originated from four prevalent types: (i) errata or course material errors, (ii) video lectures, (iii) homework, assignments or prob-

[1]We collected all threads and their component posts from four subforum categories as in Section 3.1. We did so, as we hypothesize that they would necessitate different levels of instructor intervention and that such interventions may be signaled by different features.

[2]The full list of courses is omitted here due to lack of space.

[3]As of December 2014, Coursera, a commercial MOOC platform: `https://www.coursera.org`, hosted 761 courses in English spanning 25 different subject areas.

[4]"Subforum" in Coursera terminology.

Figure 2: Coursera's forums allow threads with posts and a single level of comments.

lem sets, and (iv) exams or quizzes (see Figure 1)[5]. All 61 courses had forums for reporting errata and discussing homework and lectures. For more focused experimentation, we selected the 14 largest courses within the 61 that exhibited all four forum types (denoted "D14" hereafter, distinguished from the full "D61" dataset). Table 1 provides demographics of both D61 and D14 datasets. In our corpus, there were a total of 205,835 posts including posts and comments to posts. The Coursera platform only allows for a single level of commenting on posts (Figure 2). We note that this limits the structural information available from the forum discourse without content or lexical analysis. We observed that posts and comments have similar topics and length, perhaps the reason why previous work [18] ignored this distinction. We have retained the distinction as it helps to distinguish threads that warrant intervention.

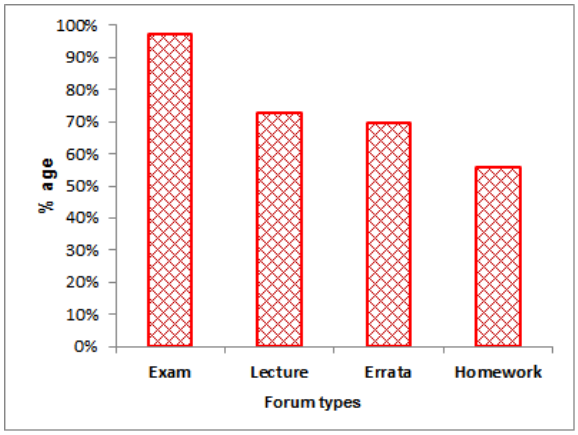


Figure 3: Thread distribution over errata, homework, lecture and exam forums in D14 by their *intervention ratio.*

3.2 System Design

From the dataset, we extract the text from the posts and comments, preserving the author information, thread structure and posting timestamps, allowing us to recreate the state of the forum at any timestamp. This is important, as we first preprocessed the dataset to remove inadmissible information. For example, since we collected the dataset after all courses were completed, instructors' posts as well as any subsequent posts in a thread need to be removed. We also do not use the number of votes or views in a thread as these are summary counts that are influenced by intervention[6].

We used regular expressions to further filter and canonicalize certain language features in the forum content. We replaced all mathematical equations by <MATH>, URLs by <URLREF> and references to time points in lecture videos by <TIMEREF>. We removed stopwords, case-folded all tokens to lowercase, and then indexed the remaining terms and computed the product of term frequency and inverse thread frequencies ($tf \times itf$) for term importance. The weighted terms form a term vector that we further normalized using the L2–norm. Other real-valued features were max–min normalized. Categorical features such as the forum type were encoded as bit vectors.

Each thread is represented as bag of features consisting of terms and specific thread metadata as disclosed below. We indicate each new feature that our study introduces with an asterisk.

1. Terms (unigrams);

2*. Forum type to which the thread belongs: Figure 3 shows a clear difference in *intervention ratio*, the ratio of number of threads intervened to those that weren't, across different forum types. Forum type thus emerges as a feature to use to discriminate threads worthy of intervention. The forum type encapsulating the thread could be one of homework, lecture, exam or errata.

3*. Number of references to course materials and other sources external to the course: includes explicit references by students to course materials within and outside the course e.g., *slide 4*, *from wikipedia*, *lecture video 7*.

4*. Affirmations by fellow students; Count of agreements made by fellow students in response to a post. Mostly, first posts in a thread receive affirmations.

5. Thread properties (Number of posts, comments, and both posts / comments, Average number of comments per post): expresses a thread's length and structural properties in terms of number of posts and comments posted.

6. Number of sentences in the thread: This feature intends to capture long focussed discussions that may be intervened more often than the rest.

7*. Number of non-lexical references to course materials: (number of URLs, references to time points in lecture videos). This feature is similar to course material references but includes only non-lexical references (Item #1) such as URLs and time points in lecture videos.

Importantly, as part of the author information, Coursera also marks instructor-intervened posts / comments. This supplies us with automatically labeled gold standard data for both training and evaluating our supervised classifier. We use threads with instructor posts / comments as positive instances (intervened threads). However, we note that the class imbalance is significant: as the instructor-student ratio is very low, typical MOOC forums have fewer positives (interventions) than negative ones. To counter skewness, we weigh

[5]Some courses had forums for projects, labs, peer assessment, discussion assignments. We omit from the collection these and other miscellaneous forums, such as those for general discussion, study groups and technical issues.

[6]Previous work such as [5] utilize this as they have access to time-stamped versions of these statistics, since they use privately-held data supplied by Coursera for MOOCs held at their university.

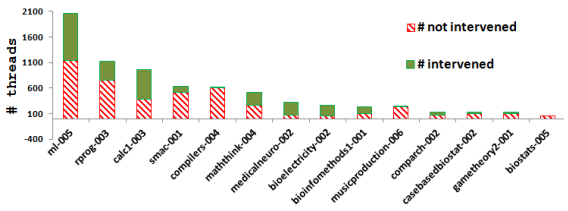


Figure 4: Thread distribution over the errata, homework, lecture and exam forums in D14. Corresponds to numeric data in Table 2.

majority class (generally positive) instances higher than minority class (generally negative) instances. These weights are important parameters of the model, and are learned by optimizing for maximum F_1 over the training / validation set.

4. EVALUATION

We performed detailed experimentation over the smaller D14 dataset to validate performance, before scaling to the D61 dataset. We describe these set of experiments in turn. As our task is binary classification, we adopted L1-regularized logistic regression as our supervised classifier in all of our experimentation.

We first investigated each of the 14 courses in D14 as 14 separate experiments. We randomly used 80% of the course's threads for training and validation (to determine the class weight parameter, W), and use the remaining 20% for testing. Our experimental design for this first part closely follows the previous work [5] for direct comparison with their work. We summarise these results in Table 2, in the columns marked "(II) Individual", averaging performance over ten-fold cross validation for each course.

The results show a wide range in prediction performance. This casts doubt on the portability of the previously published work [5]. They report a baseline performance of $F_1 \approx 25$ on both their courses each having an intervention ratio ≈ 0.13[7]. In contrast, our results show the instability of the prediction task, even when using individualized trained models. Nevertheless, on average our set of features performs better on F_1 by at least 10.15%.[8]

We observe the true intervention ratio correlates to performance, when comparing Columns I.2 and II.3 ($\rho = 0.93$). We also see that intervention ratio varies widely in our D14 dataset (Figure 4). This happens to also hold for the larger D61 dataset. In some courses, instructors intervene often (76% for medicalneuro-002) and in some other courses, there is no intervention at all (0% for biostats-005).

To see whether the variability can be mitigated by including more data, we next perform a leave-one-course-out cross validation over the 14 courses, shown in "Columns (III) LOO-course C.V.". *I.e.*, we train a model using 13 courses' data and apply the trained model to the remaining unseen test course. While not strictly comparable with (II), we feel this setting is more appropriate, as it: allows training to scale; is closer to the real scenario discussed in Section 6, Item 4.

Separately, we studied the effectiveness of our proposed set of features over the D14 dataset. Table 3 reports performance averaged over all 14 courses weighted by its proportion in the corpus. In the top half of the table, we build Systems 1–7 by cumulatively adding in features from the proposed list from Section 3.2. Although the overall result in Row 7 performs $\sim 5\%$ better than the unigram baseline, we see that the classifier worsens when the count of course references are used as a feature (Row 2). Other rows all show an additive improvement in F_1, especially the forum type and non-lexical reference features, which boost recall significantly.

[7]Based on test data figures [5] had disclosed in their work

[8]Due to non-availability of experimental data, we can only claim a 10.15% improvement over the highest F_1 they reported, 35.29.

Feature	Precision	Recall	F_1
1. Unigrams	41.98	61.39	45.58
2. (1) + Forum Type	41.36	69.13	48.01
3. (2) + Course_Ref	41.09	66.57	47.22
4. (3) + Affirmation	41.20	68.94	47.68
5. (4) + T Properties	42.99	70.54	48.86
6. (5) + Num Sents	43.08	69.88	49.77
7. (6) + Non-Lex Ref	42.37	74.11	50.56
8. (7) – Forum Type	41.33	83.35	51.16
9. (7) – Course Ref	45.96	79.12	54.79
10. (7) – Affirmation	42.59	71.76	50.34
11. (7) – T Properties	40.62	84.80	51.35
12. (7) – Num Sents	42.37	73.05	49.32
13. (7) – Non-Lex Ref	43.08	69.88	49.77

Table 3: Feature study. The top half shows performance as additional features are added to the classifier. Ablation tests where a single feature is removed from the full set (Row 7) are shown on the bottom half. Performance given as weighted macro-average over 14 courses from a leave-one-out cross course validation over D14.

The performance drop when adding in the number of course references prompts us to investigate whether removing some features from the full set would increase prediction quality. In the bottom half of Table 3, we ablate a single feature from the full set.

Results show that removing forum type, number of course references and thread length in a thread all can improve performance. Since the different rows of Table 3 are tested with weights W learnt from its own training set the changes in performance observed are due to the features and the learnt weight. When we tested the same sequence with an arbitrary constant weight we observed all features but Course_Ref improved performance although not every improvement was significant.

Using the best performing feature set as determined on the D14 experiments, we scaled our work to the larger D61 dataset. Since a leave-one-out validation of all 61 courses is time consuming we only test on the each of the 14 courses in D14 dataset while training on the remaining 60 courses from D61. We report a **weighted averaged F_1 = 50.96 (P = 42.80; R = 76.29)** which is less than row 9 of Table 3. We infer that scaling the dataset by itself doesn't improve performance since W learnt from the larger training data no longer counters the class imbalance leaving the testset with a much different class distribution than the training set.

4.1 Upper bound

The prediction results show that forum type and some of our newly-proposed features lead to significant improvements. However, we suspect the intervention decision is not entirely objective; the choice to intervene may be subjective. In particular, our work is based on

Course	(I) Demographics		(II) Individual				(III) LOO-course C.V.			
	1. # of Threads	2. I. Ratio	1. Prec.	2. Rec.	3. F_1	4. W	1. Prec.	2. Rec.	3. F_1	4. W
ml-005	2058	0.45	51.08	89.19	64.96	2.06	48.10	68.63	56.56	2.46
rprog-003	1123	0.32	50.77	48.53	49.62	2.41	35.88	75.77	48.70	2.45
calc1-003	965	0.60	60.98	44.25	51.29	0.65	65.42	72.79	68.91	2.45
smac-001	632	0.17	21.05	30.77	25.00	5.29	22.02	67.93	33.26	2.00
compilers-004	624	0.02	8.33	50.00	14.28	37.23	2.53	80.00	4.91	2.33
maththink-004	512	0.49	46.59	100.00	63.56	2.13	50.24	85.48	63.29	2.57
medicalneuro-002	323	0.76	100.00	60.47	75.36	0.32	75.86	89.07	81.94	2.34
bioelectricity-002	266	0.76	75.00	54.55	63.16	0.34	75.36	82.98	78.99	2.41
bioinfomethods1-001	235	0.55	56.00	60.87	58.33	0.78	59.67	83.72	69.68	2.36
musicproduction-006	232	0.01	0.00	0.00	0.00	185.00	0.52	50.00	1.03	2.55
comparch-002	132	0.46	47.62	100.00	64.57	1.56	48.57	83.61	61.45	2.37
casebasedbiostat-002	126	0.20	13.33	100.00	23.53	3.54	24.47	92.00	38.66	2.11
gametheory2-001	125	0.19	28.57	28.57	28.57	5.18	18.27	86.36	30.16	2.61
biostats-005	55	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	2.01
Average	529	0.36	39.95	54.80	41.59	17.68	37.64	72.74	45.54	2.36
Weighted Macro Avg	NA	0.40	45.44	61.84	49.04	10.96	42.37	74.11	50.56	2.37

Table 2: Individual course results for each course in the D14 dataset. Weights W weigh each +ve class instance w times as much as a –ve class instance. Performance varies with large variations in Intervention ratio (I-ratio) and # of threads.

the premise that correct intervention follows the historical pattern of intervention (where instructors already intervene), and may not be where general pedadogy would recommend prediction. We recognize this as a limitation of our work.

To attempt to quantify this problem, we assess whether peer instructors with general teaching background could replicate the original intervention patterns. Three human instructors[9] annotated 13 threads from the musicproduction-006 course. We chose this course to avoid bias due to background knowledge, as none of the annotators had any experience in music production. This course also had near zero interventions; none of the 13 threads in the sample were originally intervened by the instruction staff of the course.

They annotated 6 exam threads and 7 lecture threads. We found that among exam threads annotators agreed on 5 out of 6 cases. Among lecture threads at least two of three annotators always agreed. On 4 out of 7 cases, all three agreed. The apparently high agreement could be because all annotators chose to intervene only on a few threads. This corresponds to a averaged interannotator agreement of $k = 0.53$. The annotators remarked that it was difficult to make judgements, that intervention in certain cases may be arbitrary, especially when expert knowledge would be needed to judge whether factual statements made by students is incorrect (thus requiring instructor intervention to clarify). As a consequence, agreement on exam threads that had questions on exam logistics had more agreement at $k = 0.73$.

While only indicative, this reveals the subjectiveness of intervention. Replicating the ground truth intervention history may not be feasible – satisfactory performance for the task may come closer to the interannotator agreement levels: i.e., $k = 0.53$ corresponding to an F_1 of 53%. We believe this further validates the significance of the prediction improvement, as the upper bound for deciding intervention is unlikely to be 100%.

[9]The last three authors, not involved in the experimentation: two professors and a senior pedagogy researcher.

5. DISCUSSION

From handling the threads and observing discussion forum interactions across courses, several issues arise that merit a more detailed discussion. We discuss each in turn, identifying possible actions that may mollify or address these concerns. Specifically:

1. The number of threads per course varies significantly.
2. Intervention decisions may be subjective.
3. Simple baselines outperform learned systems.
4. Previous experimental results are not replicable.

Issue 1: Variation in the number of threads. We observed significant variation in the number of threads in different courses, ranging from tens to thousands. Figure 4 shows thread distribution over the D14 dataset for the errata, homework, lectures and exam forums; a similar distribution held for the larger D61 dataset. These distributions are similar to those reported earlier in the large cross-course study of [18]. The difference in number of threads across courses is due to a multitude of factors. These include number of students participating, course structure, assignment of additional credits to participating students, course difficulty, errors in course logistics and materials, etc.

When performing research that cuts across individual MOOCs, this issue becomes important. As we saw, using simple averaging on a per-course basis equates to a macro-averaging: putting each course on par in importance. However, when the decision unit is at the thread (as in our task), it makes more sense to treat individual threads at parity. In such cases, normalization at the thread level (analogous to micro-averaging) may be considered. Such thread-level normalization can affect how we weight information from each course when training in aggregate over data from multiple courses: courses with many threads should carry more weight in both training and evaluation.

Issue 2: Intervention decisions may be subjective. Instructor policy with respect to intervention can markedly differ. Instructors may only intervene in urgent cases, preferring students to do peer learning and discovery. Others may want to intervene often,

to interact with the students more and to offer a higher level of guidance. Which policy instructors adopt varies, as best practices for both standard classroom and MOOC teaching have shown both advantages and disadvantages for [12, 11].

Instructors can also manifest in different roles. In Coursera, posts and comments marked as instructor intervened can come from actual instructor intervention as well as participation by helpers, such as community teaching assistants (CTAs). We observe courses with CTAs where CTAs have a higher intervention rate. We hypothesize that such factors decreases agreements.

This plays out in our datasets. We observe that intervention is not always proportional to the number of threads in the course. Some courses such as compilers-004 (see Figure 4) has relatively fewer number of threads than other large courses. Yet its intervention rate is noticeably low. This suggests that other factors inform the intervention decision. Handling this phenomenon in cross-course studies requires an additional form of normalization.

To normalize for these different policies we can upweight (by oversampling) threads that were intervened in courses with fewer interventions. We can continue to randomly oversample a course's intervened threads until its *intervention density* reaches the dataset average. Note this normalization assumes that the few threads intervened in course with relatively low intervention density are more important; that the threads intervened for a similar high intervention density course would be a proper superset.

Even when a policy is set, intervention decisions may be subjective and non-replicable. Even with our cursory annotation of a course to determine an upper bound for intervention shows the potentially large variation in specific intervention decisions. We believe that automated systems can only approach human performance when such decisions can be subjective. As such, the upper bound for performance (cf Section 4.1) should not be construed as the single gold standard; rather, prediction performance should be calibrated to human performance levels.

Issue 3: Simple baselines outperform learned models. We also compared our work with a simple baseline that predicts all threads as needing instructor intervention. This baseline does no work – achieving 100% recall and minimal precision – but is very competitive, outperforming our learned models for courses with high levels of intervention (see Table 4). Diving deeper into the cause, we attribute this difference to the subjective nature of interventions and other extraneous reasons (bandwidth concerns) resulting in high false positive rates. That is, given two threads with similar set of features, one may be intervened while the other is not (e.g., Figure 5). This makes the ground truth and the evaluation less reliable. An alternative evaluation model might be to assign a confidence score to a prediction and evaluate the overlap between the high confidence predictions and the ground truth interventions.

Issue 4: Previous results are not replicable. From earlier work [5], intervention prediction seemed to be straightforward task where improvement can be ascribed to better feature engineering. However, as we have discovered in our datasets, the variability in instructor intervention in MOOCs is high, making the application of such previously published work to other MOOCs difficult. This is the perennial difficulty of replicating research findings. Findings from studies over a small corpus with select courses from specific subject categories may not generalise. Published findings are not verifiable due to restricted access to sensitive course data. The problem is acute for discussion forum data due to privacy and copyright considerations of students who have authored posts on those forums.

Course	Individual		D14	
	F_1	F_1@100R	F_1	F_1@100R
ml-005	64.96	63.79	72.35	61.83
rprog-003	49.62	47.39	48.55	49.31
calc1-003	51.29	74.83	70.63	75.33
smac-001	25.00	34.67	34.15	29.28
compilers-004	14.28	3.28	4.82	4.75
maththink-004	63.56	63.08	61.11	65.49
medicalneuro-002	75.36	88.66	78.06	85.67
bioelectricity-002	63.16	86.84	80.10	85.84
bioinfomethods1-001	58.33	67.65	69.40	71.07
musicproduction-006	0.00	4.35	1.09	1.72
comparch-002	64.57	55.56	60.49	63.21
casebasedbiostat-002	23.53	14.81	38.71	34.25
gametheory2-001	28.57	45.16	27.12	30.56
biostats-005	0.00	0.00	0.00	0.00
Average	41.59	46.43	45.18	47.09
Weighted Macro Avg	49.04	51.51	54.79	53.22

Table 4: Comparison of F_1 in Table 2 with those of a naïve baseline that classifies every instance as +ve – resulting in 100% recall.

The main challenge is to provision secured researcher access to the experimental data. Even in cases where researchers have access to larger datasets, such prior research [1, 5, 14, 15, 16, 22] have reported findings on each course separately (cf Table 2 "(II) Individual"), shying away from compiling them into a single dataset in their study. Bridging this gap requires cooperation among interested parties. The shared task model is one possibility: indeed, recently Rose *et al.* [17] organised a shared task organised to predict MOOC dropouts over each week of a MOOC. To effectively make MOOC research replicable, data must be shared to allow others to follow and build on published experimentation. Similar to other communities in machine learning and computational linguistics, the community of MOOC researchers can act to legislate data sharing initiatives, allowing suitably anonymized MOOC data to be shared among partner institutes.

We call for the community to seize this opportunity to make research on learning at scale more recognizable and replicable. We have gained the endorsement of Coursera to launch a data-sharing initiative with other Coursera-partnered universities. While we recognize the difficulties of sharing data from the privacy and institutional review board perspectives, we believe that impactful research will require application to a large and wide variety of courses, and that restricting access to researchers will alleviate privacy concerns.

6. A FRAMEWORK FOR INTERVENTION RESEARCH

We have started on the path of instructor intervention prediction, using the task formalism posed by previous work by Chaturvedi *et al.* [5]: the binary prediction of whether a forum discussion thread should be intervened, given its lexical contents and metadata. While we have improved on this work and have encouraging results, this binary prediction problem we have tackled is overly constrained and does not address the real-world need for intervention prediction. We outline a framework for working towards the real-world needs of instructor intervention.

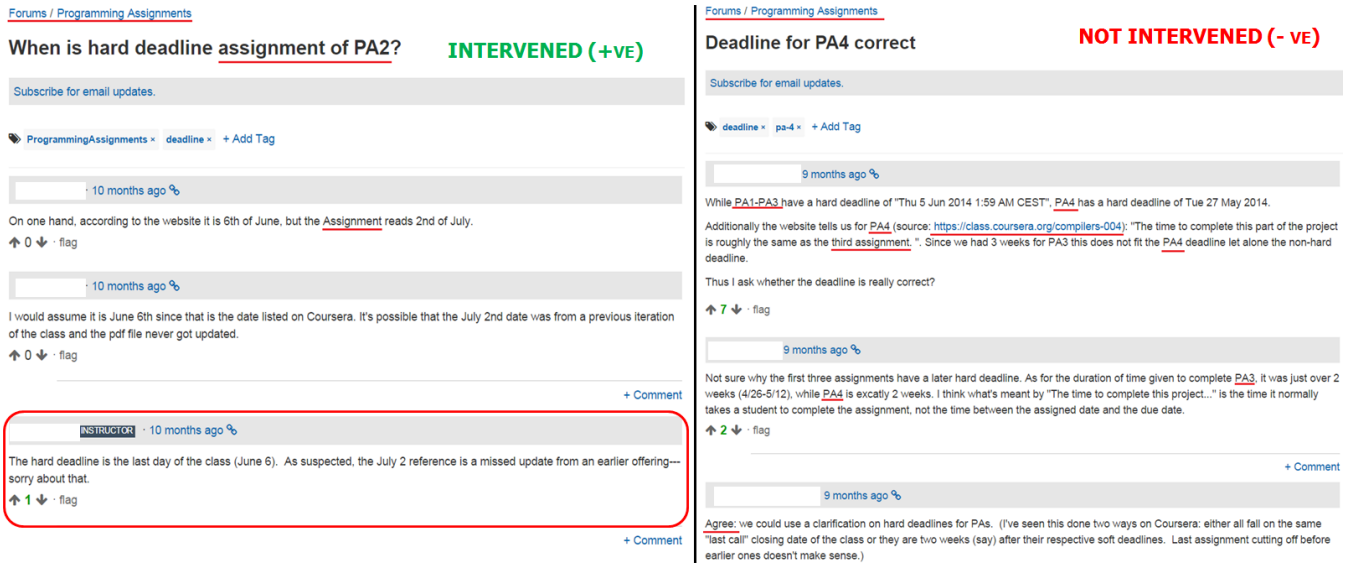


Figure 5: Interventions are, at times, arbitrary. We show two threads from `compilers-001` with similar topics, context, and features that we model (red underline). Yet only one of them is intervened (circled in red).

We thus propose a framework for investigation that iteratively relaxes our problem to take into account successively more realistic aspects of the intervention problem, with the hope of having a fieldable, scalable, real-time instructor intervention tool for use on MOOC instructors' dashboard as an end result.

1. Thread Ranking. We posit that different types of student posts may exhibit different priorities for instructors. A recommendation for intervention should also depend on thread criticality. For example, threads reporting errors in the course material may likely be perceived as critical and hence should be treated as high-priority for intervention. Even with designated errata forums, errata are reported in other forums, sometimes due to the context – *e.g.*, when a student watches a video of a lecture, it is natural for him to report an error concerning it in the lecture forum, as opposed to the proper place in the errata forum. Failure to address threads by priority could further increase the course's dropout rate, a well-known problem inherent to MOOCs [6]. Thread ranking can help to address this problem to prioritize the threads in order of urgency, which the naïve, always classifying all instances as positive, baseline system cannot perform.

2. Re-intervention. Threads can be long and several related concerns can manifest within a single thread, either by policy or by serendipity. Predicting intervention at the thread level is insufficient to address this. A recommendation for intervention has to consider not only those threads that had been newly-created but also if older threads that had already been intervened require further intervention or *reintervention*. In other words, intervention decision needs to be made in the light of newly posted content to a thread. We can change the resolution of the intervention prediction problem to one at the post level, to capture re-interventions; *i.e.*, when a new post within a thread requires further clarification or details from instructor staff.

3. Varying Teaching Roles. MOOCs require different instruction formats than the traditional course format. One evolution of the MOOC teaching format to adapt to the large scale is to recruit community teaching assistants (CTA)s. Community TAs are volunteer TAs recruited by MOOC platforms including Coursera based on their good performance in the previous iteration of the same MOOC. CTAs, traditional Teaching Assistants and technical staff are all termed as "staff" within the Coursera system. Currently, Coursera only marks threads with a "staff replied" marking, which we use directly in our training supervision in this paper. At a post level, those posted by CTAs, instructor and technical staff are marked appropriately.

We hypothesize that that these various roles differ in the quantum of time and effort, and type of content that they provide in answering posts that they contribute on a forum. It will be important to consider the role of the user while recommending threads to intervene, as the single problem of intervention may lead to n separate triaging problems for the n staff types or individual instructors that manage a MOOC.

4. Real-time. In the real world, a system needs to be predicting intervention in real-time; as new posts come into a course's forum. With ranking, we can decide when to push notifications to the instruction staff, as well as those less urgent that can be viewed at leisure on the instructor's MOOC dashboard.

With the timestamp metadata in the dataset, we have a transaction log. This allows us to easily simulate the state of a MOOC by "rewinding" the state of the MOOC at any time t, and make a prediction for a post or thread based on the current state.

This half-solves the problem. For real-world use, we also need to do online learning, by observing actual instructor intervention and adopting our system for the observed behavior. We feel this will be important to learn the instructor's intervention preference, as we have observed the variability in intervention per course, per instructor.

In our work, we focus only on the *instructor's view*, however this set of problems also has an important dual problem set: that of the *student's view*. We believe that solving both problems will have certain synergies but will differ in important ways. For example, solving the student's view will likely have a larger peer and social component than that for instructors, as MOOCs develop more social sensitivity.

7. CONCLUSION

We describe a system for predicting instructor intervention in MOOC forums. Drawing from data over many MOOC courses from a wide variety of coursework, we devise several novel features of forums that allow our system to outperform the state-of-the-art work on an average by a significant margin of 10.15%. In particular, we find that knowledge of where the thread originates from (the forum type – whether it appears in a *lecture*, *homework*, *examination* forum) alone informs the intervention decision by a large 2% margin.

While significant in its own right, our study also uncovers issues that we feel must be accounted for in future research. We have described a framework for future research on intervention, that will allow us to account for additional factors – such as temporal effects, differing instructor roles – that will result in a ranking of forum threads (or posts) to aid the instructor in best managing her time in answering questions on MOOC forums.

Crucially, we find the amount of instructor intervention is widely variable across different courses. This variability undermines the veracity of previous works and shows that what works on a small scale may not hold well in large, cross-MOOC studies. Our own results show that for many courses, simple baselines work better than supervised machine learned models when intervention ratios are high. To allow the replicability of research and to advance the field, we believe that MOOC-fielding institutions need to form a data consortium to make MOOC forum data available to researchers.

8. ACKNOWLEDGMENTS

The authors would like to thank Snigdha Chaturvedi and her co-authors for their help in answering detailed questions on the methodology of their work.

9. REFERENCES

[1] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with Massive Online Courses. In *Proc. of WWW '14*, pages 687–698. International World Wide Web Conferences Steering Committee, 2014.

[2] A. Arnt and S. Zilberstein. Learning to Perform Moderation in Online Forums. In *Proc. of WIC '03*, pages 637–641. IEEE, 2003.

[3] Y. Artzi, P. Pantel, and M. Gamon. Predicting Responses to Microblog Posts. In *Proc. of NAACL '12*, pages 602–606. Association for Computational Linguistics, 2012.

[4] L. Backstrom, J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil. Characterizing and curating conversation threads: expansion, focus, volume, re-entry. In *Proc. of WSDM '13*, pages 13–22. ACM, 2013.

[5] S. Chaturvedi, D. Goldwasser, and H. Daumé III. Predicting Instructor's Intervention in MOOC Forums. In *Proc. of ACL '14 (Volume 1: Long Papers)*, pages 1501–1511. ACL, 2014.

[6] D. Clow. MOOCs and the funnel of participation. In *Proc. of LAK '13*, pages 185–189. ACM, 2013.

[7] R. Ferguson and M. Sharples. Innovative Pedagogy at Massive Scale: Teaching and Learning in MOOCs. In *Open Learning and Teaching in Educational Communities*, pages 98–111. Springer, 2014.

[8] J. Kim, J. Li, and T. Kim. Towards identifying unresolved discussions in student online forums. In *Proc. of NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 84–91. ACL, 2010.

[9] R. Kizilcec and S. Halawa. Attrition and Achievement Gaps in Online Learning. In *Proc. of ACM L@S '15)*, Vancouver, Canada, March 14-15 2015. In Press.

[10] F.-R. Lin, L.-S. Hsieh, and F.-T. Chuang. Discovering genres of online discussion threads via text mining. *Computers & Education*, 52(2):481–495, 2009.

[11] J. Mackness, S. Mak, and R. Williams. The ideals and reality of participating in a MOOC. 2010.

[12] M. Mazzolini and S. Maddison. Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums. *Computers & Education*, 40(3):237–253, 2003.

[13] M. Mazzolini and S. Maddison. When to jump in: The role of the instructor in online discussion forums. *Computers & Education*, 49(2):193–213, 2007.

[14] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Modeling Learner Engagement in MOOCs using Probabilistic Soft Logic. In *NIPS Workshop on Data Driven Education*, 2013.

[15] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Learning Latent Engagement Patterns of Students in Online Courses. In *Proc. of AAAI '14* , 2014.

[16] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Understanding MOOC Discussion Forums using Seeded LDA. In *Proc. of 9th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–33. ACL, 2014.

[17] C. P. Rosé and G. Siemens. Shared task on prediction of dropout over time in massively open online courses. In *Proc. of EMNLP '14*, page 39, 2014.

[18] L. A. Rossi and O. Gnawali. Language Independent Analysis and Classification of Discussion Threads in Coursera MOOC Forums. In *Proc. of IEEE IRI '14)*, 2014.

[19] G. S. Stump, J. DeBoer, J. Whittinghill, and L. Breslow. Development of a framework to classify MOOC discussion forum posts: Methodology and challenges. In *NIPS Workshop on Data Driven Education*, 2013.

[20] J. H. Tomkin and D. Charlevoix. Do professors matter?: using an A/B test to evaluate the impact of instructor involvement on MOOC student outcomes. In *Proc. of ACM L@S*, pages 71–78. ACM, 2014.

[21] L. Wang, S. N. Kim, and T. Baldwin. The Utility of Discourse Structure in Identifying Resolved Threads in Technical User Forums. In *Proc. of COLING '12*, pages 2739–2756, 2012.

[22] M. Wen, D. Yang, and C. P. Rosé. Linguistic Reflections of Student Engagement in Massive Open Online Courses. In *Proc. of ICWSM '14 (poster)*, 2014.

[23] D. Yang, D. Adamson, and C. P. Rosé. Question Recommendation with Constraints for Massive Open Online Courses. In *Proc. of ACM RecSys*, pages 49–56. ACM, 2014.

Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains

Xu Wang, Diyi Yang, Miaomiao Wen, Kenneth Koedinger, Carolyn P. Rosé
School of Computer Science, Carnegie Mellon University
5000,Forbes, Pittsburgh, PA, 15213
{xuwang, diyiy, mwen}@cs.cmu.edu, koedinger@cmu.edu, cprose@cs.cmu.edu

ABSTRACT

While MOOCs undoubtedly provide valuable learning resources for students, little research in the MOOC context has sought to evaluate students' learning gains in the environment. It has been long acknowledged that conversation is a significant way for students to construct knowledge and learn. However, rather than studying learning in MOOC discussion forums, the thrust of current research in that context has been to identify factors that predict dropout. Thus, cognitively relevant student behavior in the forums has not been evaluated for its impact on cognitive processes and learning. In this paper, we adopt a content analysis approach to analyze students' cognitively relevant behaviors in a MOOC discussion forum and further explore the relationship between the quantity and quality of that participation with their learning gains. As an integral part of our approach, we built a computational model to automate the analysis so that it is possible to extend the content analysis to all communication that occurred in the MOOC. We identified significant associations between discourse behavior and learning. Theoretical and practical implications are discussed.

Keywords

Massive Open Online Courses (MOOC); Cognitive behavior; Content analysis; Discussion forum; Learning gains;

1. INTRODUCTION

Despite concerns over their effectiveness, MOOCs (Massive Open Online Courses) have attracted increasing attention both in the popular press and academia, raising questions about ptheir potential to deliver educational resources at an unprecedented scale to new populations of learners. With learning through social processes featuring among the potential impacts of MOOC platforms [5], and discussion forums currently the primary means for supporting social learning in typical MOOC platforms, recent research has begun to focus on interventions that might enrich students' interaction in this context [e.g., 30], with the purpose of providing a more engaging and effective learning experience. Previous studies on learning and tutoring systems have provided evidence that students' participation in discussion [e.g., 2, 9, 12] is correlated with their learning gains in other instructional contexts.

However, whether discussion will also contribute substantially to learning in a MOOC context, and what aspects of discussion will ultimately matter most to learning in this new context remain important open questions. Considering the significant connection that has been discovered between discussion behaviors in MOOC forums and student commitment, its potential for enabling students to form supportive relationships with other students, and the potential to enhance social learning through interaction, in depth empirical research is needed to uncover the relationship between student discourse patterns and learning gains in MOOCs.

One challenge to assessing learning in MOOCs, even in cases where formal assessments are integrated with the courses, is that students come into a MOOC with a wide variety of backgrounds [15,20], and it is typically unnatural to make a pretest a natural part of the learning process, especially when activities in the MOOC are all voluntary. However, while inconvenient, it is not impossible. The study reported in this paper took place in an unusual MOOC where a pretest was provided and students were aware that the MOOC data would be used for research purposes. This dataset, from a course entitled "Introduction to Psychology as a Science", thus provides a unique opportunity to begin to address the research questions introduced above.

Many student behaviors have been observed in discussion forums, e.g., question answering, self-introduction, complaining about difficulties and corresponding exchange of social support. A very coarse grained distinction in posts could be on vs. off topic. However, the important distinctions do not stop there and may be substantially more nuanced than that. Other than literal topic features, students' cognitively relevant behaviors, which are associated with important cognitive processes that precede learning may also be found in discussion forums. What those behaviors are in this context, and how frequently they occur are two questions we address.

Specifically, we ask the following research questions in this work:

1. Is a higher quantity of participation in MOOC discussion forums associated with higher learning gains?
2. Is on-task discourse associated with more learning gains than off-task discourse?
3. If certain properties of discussion are associated with enhanced learning, why it is so? What are the higher-order thinking behaviors demonstrated in student discourse and their connection with learning?

We consider that answering these questions has important implications for designing discussion interventions in MOOCs.

Some previous studies on MOOC discussion forums analyzed at a macro-level the quantity of participation [e.g., 1], whereas other work [23] pointed out that quantitative indices of participation does not directly imply the quality of conversation and interaction. Others conducted content analysis of thread topics [17] or used rule-based algorithms to extract linguistic markers [28]. However, students' higher-order thinking behaviors are not well represented

or thoroughly and systematically explored in these previous investigations. In this work, we aim to adopt a content analysis approach to hand-code data based on a well-established learning activity classification framework from earlier cognitive science research [8] in an attempt to capture students' discussion behaviors and their underlying cognitive strategies in a MOOC discussion forum. This is the first work we know of that has brought this lens to explore students' discussion behaviors and their association with learning gains in MOOCs.

In particular, we contribute to the existing literature by 1) developing a coding scheme based on Chi's ICAP (Interactive-Constructive-Active-Passive) framework [8] in categorizing students' discussion behaviors in a MOOC context; 2) providing empirical support for the importance of discussion in enhancing learning in a MOOC context. We also contribute to the literature on computer-supported collaborative learning by exploring the relationship between discourse and learning in a multi-user distributed asynchronous discussion environment.

In the remainder of the paper, we first discuss related work and existing theoretical foundations that we leverage in our analysis. Next we introduce our dataset. We then describe our methods, including specifics about the coding scheme, and computational model in the Methods section. We present an extensive correlational analysis and then discuss our interpretation along with caveats and directions for continued work.

2. RELATED WORK

2.1 Research on MOOC discussion forums

Studies in the field of learning science and computer supported collaborative learning have provided evidence that learners' contribution to discourse is an important predictor of their knowledge construction [2, 12]. In offline environments, studies have suggested, for example, that the number of words per utterance [26] and proportion of words produced [14] are correlated with learning gains. Transitioning from traditional classroom to online learning, computer-mediated conferencing has proved to be a gold mine of information concerning students' psycho-social dynamics and their knowledge acquisition [19]. Investigating the usage of discussion forums in MOOCs has been one major theme for research. To give a few examples, at a participation level, Anderson and colleagues [1] found that students who participated in other platform activities (videos, quizzes, etc.) participated more in the forum as well. They also explored patterns of thread initiators and contributors in terms of specific discussion behaviors in the discussion forum. At a content level, Brinton [5] categorized discussion threads into "small-talk", "course logistics", and "course specific" categories. Gillani [17] adopted a content analysis approach combined with machine learning models to discover sub-communities in a MOOC based on user profiles. Anderson [1] used a lexical analysis to see which words predict the number of assignments a student finally turns in.

These studies have set up a good foundation for analyses in MOOC discussion forums. However, to confirm a relationship between discussion and learning, we need to look closer into what aspects of discussion actually contribute to learning from a cognitive perspective.

2.2 Content analysis

We base our work on previous approaches to analyze content of student dialogues in tutoring and computer-supported collaborative learning environments. Chi [6] pointed out the importance of verbal analysis, which is a way to indirectly view student cognitive activity. De Wever [16] further demonstrated that content analysis has the potential to reveal deep insights about psychological processes that are not situated at the surface of internalized collaboration scripts.

Chi's ICAP framework [8] has been considered to be the strongest evidence for the value of a dialogic approach to learning [25], which has been widely adapted and applied to identify learning activities and explain study results [e.g., 24, 27]. The framework has been utilized to explain classical educational experiments [10] and serve as a theoretical foundation for studies on tutoring and computer-supported collaborative learning, for example in a discourse analysis of different kinds of scaffolds [24].

The framework was created through a meta-analysis of 18 studies in which learning activities were classified into 3 categories, namely, interactive activities that involve discussing and co-constructing with a peer or the learning environment, constructive activities that produce a representation of information that goes beyond the presented information, and finally, active activities that show how students are actively engaged in the learning process. The taxonomy suggests the hypothesis that what are referred to in it as interactive activities should generate more learning outcomes than constructive activities, which in turn should generate more learning outcomes than active activities. [8]

MOOCs provide an emerging environment where computer-supported collaborative learning activities might be provided, and where social presence might reflect cognitive presence [27]. Thus, in this context we aim to apply the ICAP framework to explore the relationship between discussion and learning by coding observed student behaviors in the discussion forum.

3. DATASET

In this work, we conducted a secondary analysis of the dataset of the course "Introduction to Psychology as a Science" offered through Coursera collaboratively by Georgia Institute of Technology and Carnegie Mellon University. The course incorporated elements of the OLI (Open Learning Initiative) "Introduction to Psychology" learning environment. One special characteristic of the course was that it administered a pre/post test with the intention to support research.

"Introduction to Psychology as a Science" was designed as a 12-week introductory course. For each week of class, the course targeted a major topic (e.g. Memory, Brain Structures, Nervous System); Course materials include video lectures, assigned MOOC activities, learning activities in the OLI environment, and what are referred to as weekly high-stakes quizzes.

In the first analysis of the dataset [21], researchers found that students who registered for the OLI activities learned more than students who used only the typical MOOC affordances, and further demonstrated that students who did more learning-by-doing activities learn more than students who watch more videos or read more texts. In other words, doing an activity has a much greater effect (6x) on predicted learning outcomes than watching a video or reading a web page. However, students' participation in the discussion forum hasn't been explored yet in that work.

In our preliminary exploration into the dataset, we found that when controlling for students' registration for OLI activities (which serves as a control variable associated with effort and commitment to the course), their quantity of participation in discussion forums significantly predicts learning gains as well. Based on this, we wanted to further explore how students' specific cognitively relevant behaviors in the forums correlate with their learning gains. We observed specific related discourse behaviors in the forum, and present several examples here.

Active behavior: *"According to the OLI textbook, creative intelligence is 'the ability to adapt to new situations and create new ideas or practicality'."*

This is an example of the student actively repeating what's being said in the course materials.

Constructive behavior: *"When I tell my son to wash the dishes, it's much more straightforward to explain his refusal or agreement by some behavioral (e.g. Reward or punishment) or cognitive mechanisms than by an innate instinct to wash or not to wash the dishes."*

This is an example of constructive behavior, when the learner produces output, which could be examples, explanations, etc., that go beyond course materials.

Interactive behavior: *"I agree that language can be an extra difficulty, but it is not a variable with which is counted. Also, depression, work stress…could form extra difficulties for the student in particular."*

This interactive behavior example shows that students not only engage in self-construction, but build their ideas upon their partners' contributions.

Altogether, there are 27,750 registered users in the dataset, and 7,990 posts and comments in the dataset. For the learners who have both pretest and posttest on record, which is our population of interest, there are 3,864 posts in total and 491 users. In addition to forum records, student clicks with course materials are also recorded in the clickstream data. The course has 1,487,665 student clicks. The clickstream logfile provides us with the opportunity to observe each students' interaction with course materials.

4. METHOD

4.1 Unit of analysis

In this paper, our unit of analysis is the message. As proposed in [16], in their review of 15 instruments in doing content analysis of the transcripts of online asynchronous discussion groups, 7 recommended using the message as the unit of analysis.

We first looked at students' quantity of participation, and distinguished on-task discourses from off-task. We then applied a coding scheme on on-task discourse to capture the cognitive behaviors in the discussion forum. We hand-coded half of the dataset, and trained a machine learning model to replicate that annotation approach in the rest of the dataset.

In a MOOC context, the data we usually deal with is student log data [4, 5, 13], which illustrates their participation process. However, students' cognitive behaviors are better represented in their discourse displayed in the discussion forum. In this work, we hand-coded a large sample of the dataset, which may reduce noise in this kind of analysis. Thus the result may be more reliable in demonstrating the relationship between students' cognitive behaviors in the discussion forum and their learning gains.

4.2 Quantity of participation

H1: In response to our first research question, we hypothesized that students who participated more in the discussion forum have higher learning gains.

We quantified students' participation in the discussion forum by the variable PostCountByUser.

PostCountByUser: It is measured by the number of posts a user posted in the discussion forum.

We did not distinguish between posts and comments in this analysis. So the word posts when mentioned in the rest of the paper refers both to posts and comments.

4.3 On-task vs. Off-task discourse

H2: in response to our second research question, we distinguished on-task and off-task discourse in the dataset. And we hypothesized that students' total number of on-task discourse contributions has a positive association with their learning gains.

We distinguished on-task discourse from off-task discourse in the dataset, based on the following definitions. On-task discourse includes posts that talk about course content, the content of quizzes and assignments, comments on course materials, and interaction between students on course content-related issues. Off-task discourse includes posts that talk about administrative issues in the course, e.g., asking for extensions on assignments; technical issues regarding course materials, e.g., asking where to download videos, off-topic self-introductions and social networking.

This feature in the dataset is acquired through hand-coding.

4.4 Cognitively Relevant Discussion behavior

H3: In response to our third research question, we want to investigate what discussion behaviors are demonstrated in the discussion forum, their frequencies and their association with learning. In order to capture these discussion behaviors, we developed a coding scheme based on Chi's ICAP framework [8].

We further hypothesized that students who demonstrated more higher-order thinking behaviors in each of the categories, active discourse, constructive discourse, and interactive discourse have higher learning gains. And according to Chi's work represented in 18 empirical studies, we hypothesized that the effect follows the pattern interactive>constructive>active.

4.4.1 Coding scheme

Students' cognitive behaviors are reflected in the MOOC discussion forums, which is not easily mined through rule-based algorithms due to its scale and informal style. This may pose challenges for computational modeling. In this work, we adopt a hand-coding method to capture higher-order thinking behaviors and follow the hand coding with computational modeling.

Within the category of on-task discourse we divide all posts into 3x3 categories as listed in Table 1 according to Chi's Active-Constructive-Interactive framework [8]. We further offer operational definitions for each category, and provide examples from our dataset. Due to space limitations, we provide abbreviated definitions rather than the full ones provided to the human coders. When defining each category of cognitive behavior, we evaluated how this might contribute to learning. Through empirical observation, we found this coding scheme to be exhaustive of all conditions. The 9 categories are not mutually exclusive. Thus, a post may belong to more than one of these fine-grained categories.

4.4.2 Inter-rater reliability

Two experts separately coded 100 posts randomly selected from the dataset, and applied on- vs. off-task annotation plus the 9 fine-grained categories of discussion behaviors to the sample. The two experts at first reached an agreement statistic of 0.52 (Cohen's Kappa), which is a moderate level of agreement. The two experts then resolved their disagreements through consensus coding by discussing and clarifying some borderline cases. After higher consensus was achieved, one of the experts coded 2000 posts randomly sampled from the whole dataset (3864 posts).

Table 1. Coding Examples

Active Discourse- (1) Repeat	
Operational Definition: The learner explicitly repeats information that's already covered in the material, which could be indicated by quotes.	*E.g. 1: Week 2, I quote from the picture: "The portion of the sensory and motor cortex devoted … as does the entire trunk of the body."*
Active Discourse- (2) Paraphrase	
Operational Definition: The learner paraphrases what's covered in course materials, it could be indicated by words like "it's mentioned in the textbook…", "it's said in the video…"	*E.g. 2: On the chapter about Health Psychology there is a board depicting various factors about Happiness, such as the Inequality of Happiness and then the Inequality Adjusted Happiness.*
Active Discourse- (3) Notes-taking	
Operational Definition: The learner mentions about note-taking and information seeking.	*E.g. 3: I use the text files as a basis for my lecture notes.*
Constructive Discourse- (1) Ask novel questions	
Operational Definition: The learner proposes a novel question or problem based on his/her own understanding.	*E.g. 4: Violence is throughout our history and have shaped societies, is it really as simple as an observed response? or a throwback of survival instinct?*
Constructive Discourse- (2) Justify or provide reasons	
Operational Definition: The learner uses examples and evidence to support a claim he/she has made. Reasoning is explicitly demonstrated in the discourse.	*E.g. 5: It depends on the visual field. Signals from the right visual field come to the left hemisphere, while signals from the left visual field come to the right hemisphere.*
Constructive Discourse- (3) Compare or connect	
Operational Definition: The learner compares cases, connects or shares links to external resources.	*E.g. 6: Here's a link to an article about a lady who stopped dreaming after suffering a stroke: [link]*
Interactive discourse- (1) Acknowledgement of partners' contribution	
Operational Definition: The learner explicitly acknowledges their partners' contribution, which could be indicated by "thanks for pointing that out", "I agree with you there…".	*E.g. 7: That's an interesting point, and it has made me wonder why this example was chosen.*
Interactive discourse- (2) Build on partners' contribution	
Operational Definition: The learner makes a point that builds on what their partner has said.	*E.g. 8: I do agree with what you said to a large degree. Changing a behavior merely to avoid pain or any other form of punishment is not good… Hence it requires a much deeper introspection and understanding…*
Interactive discourse- (3) Defend and challenge	
Operational Definition: The learner challenges his/her partners' ideas, or defends their own ideas, when there is a disagreement. (Note: The partner here can be either a peer or the learning environment)	*E.g. 9: I think I understand what you mean (I am currently doing the statistics course as well). However, as I can see from what you've described, you still have the hypothesis in your psychological experiment which is not null - your prediction that something WILL happen.*

4.4.3 Computational model and data preparation

In order to better visualize the dataset and potentially apply the model to another context, we trained a computational model based on the coded 2000 posts to predict the cognitively relevant discussion behavior categories and expand the coding to the rest of the dataset.

In our hand-coded dataset, we labeled 9 types of cognitively relevant discussion behaviors, but due to the fact that the occurrences of each single category are relatively sparse, we acquired a low accuracy when using the sample to train a model and apply it to the rest of the dataset. Instead, we aggregated the 9 categories into the three major categories—Active, Constructive, and Interactive. All three are binary variables indicating whether the user has a post under this category. We then built models to predict these labels.

Our classifier is designed to predict whether the cognitive behavior expressed in a post belongs to Active (A), Constructive (C) or Interactive (I) by taking advantage of a bag-of-words representation. However, we have to distinguish between on-task discourse and off-task discourse since learning relevant cognitive behaviors will occur primarily in on-task discussion (Among our coded 2000 posts, 558 are on-task discussions).

For this purpose, we built a two-stage classification model. In the first stage, we designed a logistic regression classifier to predict whether a post is on-task or off-task; in the second stage, we classified the posts that were predicted to be on-task into A, C or I categories. The input for each classifier is a bag-of-words feature representation. In the first step, we used the coded 2000 posts as the training set to train a logistic regression classifier to distinguish on-task discourses and off-task discourse, and in the second step, we used 588 on-task messages as our training set to train three logistic regression classifiers to label on-task

discourses in the three categories (A, C, I). On the training set, we adopted a 10-fold cross-validation approach to evaluate the model. The classification results presented in Table 2 are the average accuracy and Kappa for this cross-validation. The results show that both accuracy and kappa are within a reasonable range for our further analysis of the whole dataset.

Table 3 shows some top-ranked features identified by the classifiers that are used to predict the three cognitive behaviors. From this table, we can see that in active discourse, students more often mentioned "lectures" "page" "notes", which indicates they're actively engaged with the course materials. In constructive discourse, students more often mentioned words associated with reasoning, such as "more" "but" "hence" "examples", and in interactive discourses, students mentioned "your" "agree" "disagree" more often, which implies interaction. These features are consistent with our initial definitions of these distinct categories of discussion behavior and assumptions about their underlying cognitive processes and strategies.

Table 2. Evaluation metrics of the computational model.

Evaluation Metrics		Accuracy	Kappa
1st Stage **On-task Prediction**	On-task	82.1%	0.527
2nd Stage **Cognitive Behavior**	Active	74.3%	0.361
	Constructive	75.4%	0.318
	Interactive	75.6%	0.236

Table 3. Performance of Discussion Behaviors Regressors and Top Ranked Features

Categories	Active	Constructive	Interactive
Most Important Word Features (Regression Weight)	***lecture (1.68) page (1.24) what (.84) text (.83) incorrect (.79) answer (.72) says (.72) notes (.68)***	***course (.87) more (.79) give (.75) trying (.68) but (.64) hence (.64) looking (.61) topics (.58) example (.56) because (.56)***	***your (1.56) agree (1.11) our (.99) again (.86) thanks (.76) disagree (.6) response (.6)***

4.4.4 Clickstream Data

In order to explore the relationship between cognitively relevant discussion behaviors and learning, we also need to control for students' involvement in other activities in the MOOC environment other than the discussion forum. This enables us to isolate, to some extent, the effect of pure effort and engagement in the course from the effects specifically related to discussion behavior. We further generated the following control variables through mining clickstream data of the course.

Video: The variable was computed first by summing the number of unique videos the student started to watch (Based on clicks on unique video urls), and then standardizing the sums.

Quiz: The variable was computed first by summing the number of unique quizzes the student attempted (Based on clicks on unique quiz urls), and then standardizing the sums.

OLI_textbook: The variable indicates reading the OLI textbook, and it's calculated by summing the number of clicks the student made in the OLI environment and then standardizing the sums.

5. RESULTS

5.1 Participation quantity in the discussion forum

In response to the first research question, we fitted linear regression models to explore the relationship between students' quantity of participation and their learning gains.

In the dataset, there are 1,079 students out of 27,750 students (i.e., students who registered for the course) who have both pre- and post-test scores on record. And among these students, there are 491 students who have posted in the discussion forum, with a total of 3,864 posts. We now introduce the variables we used in these models.

Dependent variable:

Post-test: The dependent variable in all the following models is students' posttest score, which is standardized. Post-test score is students' final exam score composed of 35 multiple-choice questions.

Control variable:

Pre-test: This is a test students took before the course started, which contains 20 multiple-choice questions. We also standardized the pretest score.

OLI_Registration: This is a binary variable capturing whether the student has registered for OLI or not. 1 means the student registered for OLI. As demonstrated in [21], students who registered for OLI learnt more than students who didn't.

We also controlled for students' involvement in other activities, including Video, Quiz and OLI_textbook.

Independent variable:

Participation: This is a binary variable indicating whether the student has ever posted in the discussion forum during the course.

PostCountByUser: This is the total number of posts a student contributed in the discussion forum during the course.

From Model 1, we see that whether the student has participated in the discussion forum is a significant predictor of the student's learning gains. The result from Model 2 shows that for those who have participated in the discussion forum, the more they posted, the higher the learning gains they achieved.

Table 4. Regression results of learning gains on the quantity of participation and on-task discourse

Control/Indep. Variable	Model 1 (N=1079)	Model 2 (N=491)	Model 3 (N=491)
Participation	0.089**		
PostCountByUser		0.005*	0.006*
OnTaskPercent.			0.123**
Pretest	0.196***	0.254***	0.243***
OLI_registration	0.119**	0.107	0.120
Video	0.056*	0.0001	-0.011
Quiz	-0.008	-0.035	-0.037
OLI_textbook	0.050**	0.048	0.044

(p<0.001***, p<0.01**, p<0.05*)

5.2 On-task versus off-tasks discourse

In response to the second research question, we looked at whether students' on-task discourse contributes to their learning gains. In this model, we examine the main effect of on-task discourse, which is represented by the variable OnTaskPercent.

Independent variable:

OnTaskPercent. : This is measured by the number of a student's posts that are categorized as on-task divided by the total number of posts the student has made, and the value is standardized.

In this regression model, we also controlled for a student's number of posts, whether they registered for OLI, pretest score, and their involvement in other activities. The regression result is displayed in Table 4-Model 3. The result shows that the quantity of students' on-task discourse in the discussion forum is a significant predictor of their learning gains.

5.3 Cognitively relevant discussion behavior analysis

5.3.1 Active, Constructive and Interactive behaviors

In this section we examine the relationship between students' discussion behavior and their learning gains and attempt to explain why certain behaviors lead to learning. We built linear regression models to explore the relationship between students' active, constructive and interactive discussion behaviors and their learning gains.

In the whole dataset, the number of instances (N=3864) of active, constructive and interactive activities is respectively 269, 744 and 203. And the number of students (N=491) who have demonstrated active, constructive and interactive activities is respectively 114, 230 and 84.

Our independent variables include:

Active, Constructive, Interactive: All three are binary variables indicating whether the student has a post that is categorized in that category.

We also controlled for variables including pretest, the number of posts, whether registered for OLI, students' involvement in other activities, as defined above. The regression result is shown in Table 5.

In Model 4 and Model 5, we found that students who have demonstrated active and constructive behaviors in the discussion forum had significantly more learning gains than students who didn't. From Model 6, we can see that the effect of Interactive discussion behavior is not significant in predicting learning gains. And we then introduced another variable to define whether a user is an active poster by counting the total number of their posts.

Poster profile: This nominal variable categorizing users into active poster and inactive poster. If a user has more than 3 posts (including 3), he/she is categorized as an active poster, otherwise categorized as an inactive poster. 3 is the median of the number of posts.

When nesting interactive behaviors with a poster profile, we found that interactive discussion is a significant predictor of learning gains for students who posted less. We think this might be because the number of posts is a basic measure of a student's social engagement in the discussion forum, which overlaps with some behaviors under the Interactive category. We further fitted a regression model to check the correlation between a student's total number of posts and the number of posts that are categorized as Interactive. The result shows that Interactive posts account for 66% of the variance in the total number of posts. We consider this high correlation could lead to the result described above. The results here show that both active and constructive discussion behaviors significantly contribute to students' learning gains, with active behaviors having higher predictive power. For users who posted less in the discussion forum, interactive behaviors strongly predict their learning gains (coefficient=0.515), however, the effect of interactive behavior disappears on the overall user population.

In addition to the occurrence of different discussion behaviors, we also used the frequency of each behavior as independent variables and did a second round of regression, from which we acquired similar results.

Table 5. Regression results of learning gains on discussion behaviors (part 1, N=491)

Control/Indep. Variable	Model 4	Model 5	Model 6	Model 7
Active	0.125*			
Constructive		0.112*		
Interactive			0.106	
Interactive [inact._poster]				0.496*
Interactive [act._poster]				0.043
Pretest	0.252***	0.246***	0.254***	0.254***
PostCntByUser	0.004	0.004	0.004	0.004
OLI_registr.	0.125	0.109	0.104	0.115
Video	-0.004	0.015	0.003	0.007
Quiz	-0.039	0.036	-0.038	-0.036
OLI_textbook	0.034	0.044	0.040	0.036

(p<0.001***, p<0.01**, p<0.05*)

5.3.2 Specific discussion behaviors

From the hand-coded dataset (N=2000), we summarized the occurrences of the 9 sub-categories of behaviors in Table 6. It shows that the most frequent behavior in the discussion forum is proposing an idea or asking novel questions. And the least frequent behaviors include building on a partner's contribution as well as defending and arguing, which is consistent with our expectation that higher-order thinking behaviors and highly interactive behaviors are relatively rare in MOOC discussion forums, and that the conversations going on in MOOCs are not satisfactorily rich and interactive.

Table 6. Distribution of 9 categories of discussion behaviors

Behavior Type	Freq.	Behavior Type	Freq.
Repeat	53	Notes-taking	28
Paraphrase/ask shallow questions	103	Justify or provide reasons	118
Propose an idea/ask novel questions	315	Compare, connect/ Reflect	59
Acknowledge partners' contribution	101	Build on partners' contribution	23
Defend and argue	14		

We also fitted regression models on this more nuanced coded dataset, but due to the fact that the occurrences of each category is relatively sparse, there was not sufficient statistical power to detect a significant effect of every category on learning gains. We display just the 2 significant predictors (out of 9) in Table 7.

Independent variables:

Constructive-(1): This is a binary variable indicating whether the student has a post that is categorized as "propose an idea/ ask novel questions".

Constructive-(2): This is a binary variable indicating whether the student has a post that is categorized as "Justify or provide reasons".

We controlled for pretest, number of posts, and whether the student registered for OLI in the regression models. We also controlled for students' involvement in other activities, the effects of which aren't significant in the regression models, so we don't report them here in Table 7.

Table 7. Regression results of learning gains on discussion behaviors (part 2, N=399)

	Model 8	Model 9
Constructive-(1)	0.136*	
Constructive-(2)		0.211**
Pretest	0.205***	0.198***
OLI_registration	0.225	0.214
Number of posts	0.007	0.005

($p<0.001$***, $p<0.01$**, $p<0.05$*)

After fitting regression models of learning outcome and all discussion behaviors as main effects, we found that only two categories are significant in predicting learning gains, as shown in Table 7. We consider higher frequencies of both behaviors might be the reason leading to significant effects on learning. Nevertheless, the processes of proposing an idea or a problem, and providing examples and reasons to justify a claim are considered to be higher-order thinking behaviors that have been proved to be instrumental to learning in several studies [e.g., 7, 18, 22], which could also lead to the significant effects.

6. CONCLUSION AND DISCUSSION

In this paper we adopted a content analysis approach and developed a coding scheme to analyze students' discussion behaviors, which are hypothesized as relating to their underlying cognitive processes in the discussion forum of a MOOC. The learning gains measures available for students in this MOOC enabled us to explore the relationship between students' discussion behaviors and their learning, and discuss what aspects of discussion appear to contribute to learning.

We observed that students' active and constructive discussion behaviors are significant in predicting students' learning gains, with active discussion behaviors possessing better predictive power, which is inconsistent with our hypotheses. Interactive discussion behaviors are significant in predicting learning gains only for students who are less active in the forums. This work also provides insight into how students are discussing in the discussion forum now, what behaviors they demonstrate and what the underlying cognitive processes are.

6.1 Active-Constructive-Interactive framework

Based on Chi's framework [8], we hypothesized that students' interactive discussion behaviors will produce more learning gains than constructive behaviors, and constructive behaviors will produce more learning gains than active behaviors. However, in this analysis we found that students' active discussion behaviors are most effective in predicting students' learning gains (coefficient=0.125). In our categorization of active behavior, students are talking about what is already covered in the materials, repeating statements that had appeared in the textbook or video lectures, and asking clarifying questions about definitions, implicitly expressing confusion about course materials, etc. According to Chi's framework [8], constructive activities should provide better learning outcomes than active activities. An example of this is when students need to explain in a constructive condition. However, we consider one reason we may not have seen this pattern in our dataset is that the post-test may not have targeted the skills and concepts students learned from these constructive activities. Assessments of a different nature, for example incorporating more demanding open ended response items, may have been more sensitive to these gains. For example, when the learning task is about design of psychology experiments, an assessment of requiring the students to design an actual experiment might be more telling than multiple-choice questions in measuring students higher-order thinking skills.

6.2 Invisible learning practices

In this paper, we looked at students' overt discussion activities in the forum, however students may be engaged in these higher order thinking activities without articulating their reasoning in a visible discourse. As indicated by [3], reading but not necessarily posting can be a productive practice for some learners. Our estimates of the amount of videos, quizzes and OLI textbook pages attempted could also be improved, for example, using the time spent on each activity, and further details about the attempt of OLI activities could be incorporated, as defined and estimated in [21].

6.3 Design implications

As MOOCs evolve, our focus as a community will transition from a primary concern about retaining users to actively improving the pedagogical effectiveness of this learning environment. Thus we need an empirical foundation to base designs for discussion affordances in MOOCs that might facilitate constructive and interactive conversations. Also, we need to come up with better assessment methods to assess and acknowledge students' higher-order thinking behaviors and skills they acquired through reading others' ideas, explaining and arguing in a discussion forum.

The paper proposes a manual way to hand-code students discussion behaviors, and offers a machine learning model to predict the corresponding behaviors in all communications of the dataset. We haven't had the opportunity to test the model in other courses, as few courses have pre- and post-test measures. If the computational model can be applied, we may provide feedback on students' advanced discussion behaviors in the forum, in terms of their cognitive processes and strategies.

7. ACKNOWLEDGEMENTS

This research was funded in part from NSF DATANET grant 1443068 and a grant from Google.

8. REFERENCES

[1] Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2014). Engaging with massive online courses. *In Proceedings of the 23rd international conference on World wide web* (pp. 687-698). International World Wide Web Conferences Steering Committee.

[2] Barab, S., & Duffy, T. (2000). From practice fields to communities of practice. In D. H. Jonassen & S.M. Land (Eds.), *Theoretical Foundations of Learning Environemnts*.

[3] Beaudoin, M. F. (2002). Learning or lurking?: Tracking the "invisible" online student. *The internet and higher education*, 5(2), 147-155.

[4] Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8(1), 13-25.

[5] Brinton, C., Chiang, M., Jain, S., Lam, H., Liu, Z., & Wong, F. (2013). Learning about social learning in MOOCs: From statistical analysis to generative model, *7*(4), 346–359.

[6] Chi, M. T. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The journal of the learning sciences*, 6(3), 271-315.

[7] Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In R. Glaser (Ed.), *Advances in instructional psychology* (pp. 161–238). Mahwah, NJ: Erlbaum.

[8] Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1), 73-105.

[9] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T.,&Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, 25, 471–533.

[10] Chi, M. T. H., & VanLehn, K. a. (2012). Seeing Deep Structure From the Interactions of Surface Features. *Educational Psychologist*, *47*(3), 177–188.

[11] Clow, D. (2013). MOOCs and the funnel of participation. *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp185-189). ACM.

[12] Cohen, E. G. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, 64, 1–35.

[13] Coffrin, C., Corrin, L., de Barba, P., & Kennedy, G. (2014). Visualizing patterns of student engagement and performance in MOOCs. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge - LAK '14* (pp. 83–92). New York, New York, USA: ACM Press.

[14] Core, M. G., Moore, J. D., & Zinn, C. (2003). The role of initiative in tutorial dialogue. *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (pp. 67–74). Morristown, NJ: Association of Computation Linguistics.

[15] DeBoer, Jennifer, G. S. Stump, D. Seaton, and Lori Breslow. (2013). Diversity in MOOC students' backgrounds and behaviors in relationship to performance in 6.002x. *In Proceedings of the Sixth Learning International Networks Consortium Conference*.

[16] De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education*, *46*(1), 6–28.

[17] Gillani, N., Eynon, R., Osborne, M., Hjorth, I., & Roberts, S. (2014). Communication communities in MOOCs. *arXiv preprint arXiv:1403.4640.*

[18] Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American educational research journal*, 31(1), 104-137.

[19] Henri, F. (1992). Computer conferencing and content analysis. *In Collaborative learning through computer conferencing* (pp. 117-136). Springer Berlin Heidelberg.

[20] Kizilcec, R. F., Piech, C., & Schneider, E. (2013). Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. *In Proceedings of the third international conference on learning analytics and knowledge* (pp. 170-179). ACM.

[21] Koedinger, K.R., Kim, J., Jia Z., McLaughlin E., Bier, N. (2015). Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC. *Learning at Scale '15*.

[22] Mestre, J. P. (2001). Implications of research on learning for the education of prospective science and physics teachers †. *Physics Education*, *36*(1), 44–51.

[23] Meyer, K. (2004). Evaluating online discussions: four different frames of analysis. *Journal of Asynchronous Learning Networks*, 8(2), 101–114.

[24] Molenaar, I., Chiu, M. M., Sleegers, P., & van Boxtel, C. (2011). Scaffolding of small groups' metacognitive activities with an avatar. *International Journal of Computer-Supported Collaborative Learning*, *6*(4), 601–624.

[25] Osborne, J., Simon, S., Christodoulou, A., Howell-Richardson, C., & Richardson, K. (2013). Learning to argue: A study of four schools and their attempt to develop the use of argumentation as a common instructional practice and its impact on students. *Journal of Research in Science Teaching*, *50*(3), 315–347.

[26] Rosé, C. P., Bhembe, D., Siler, S., Srivastava, R., & VanLehn, K. (2003). Exploring the effectiveness of knowledge construction dialogues. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Artificial intelligence in education: Shaping the future of learning through intelligent technologies* (pp. 497–499). Amsterdam, the Netherlands: IOS Press.

[27] Shea, P., & Bidjerano, T. (2012). Learning presence as a moderator in the community of inquiry model. *Computers & Education*, 59(2), 316–326.

[28] Wen, M., Yang, D., & Rose, C. P. (2014, May). Linguistic reflections of student engagement in massive open online courses. *In Proceedings of the International Conference on Weblogs and Social Media*.

[29] Wen, M., Yang, D., & Rosé, C. P. (2014). Sentiment Analysis in MOOC Discussion Forums: What does it tell us. *Proceedings of Educational Data Mining.*

[30] Yang, D., Adamson, D., & Rosé, C. P. (2014). Question recommendation with constraints for massive open online courses. *Proceedings of the 8th ACM Conference on Recommender Systems - RecSys '14*, 49–56.

[31] Zimmerman, B. J. (1989). A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology*, 81(3), 329–339.

Methodological Challenges in the Analysis of MOOC Data for Exploring the Relationship between Discussion Forum Views and Learning Outcomes

Yoav Bergner
Educational Testing Service
Princeton, NJ 08541
ybergner@ets.org

Deirdre Kerr
Educational Testing Service
Princeton, NJ 08541
dkerr@ets.org

David E. Pritchard
M.I.T.
Cambridge, MA 02139
dpritch@mit.edu

ABSTRACT
Determining how learners use MOOCs effectively is critical to providing feedback to instructors, schools, and policy-makers on this highly scalable technology. However, drawing inferences about student learning outcomes in MOOCs has proven to be quite difficult due to large amounts of missing data (of various kinds) and to the diverse population of MOOC participants. Thus significant methodological challenges must be addressed before seemingly straightforward substantive questions can be answered. The present study considers modeling final exam performance outcomes on early-stage ability estimates, discussion forum viewing frequency, and overall assessment-oriented engagement (AOE, seen as a proxy measure of motivation). These variables require careful operationalization, analysis of which is the principle contribution of this work. This study demonstrates that the effect sizes of discussion forum viewing activities on final exam outcomes are quite sensitive to these choices.

Author Keywords
MOOCs; discussion forums; social learning.

INTRODUCTION

Massive open online courses (MOOCs), a recent modality of distance learning wherein course materials are made available online and are freely accessible by anyone with computer access, have been rapidly gaining popularity as new platforms and courses come online. As of August 2014, over 2000 MOOCs were being offered through more than 50 initiatives (www.mooc-list.com), and these numbers had more than doubled over the prior year. MOOCs are generally viewed as having great value because they provide expanded opportunities to learn and near-instantaneous feedback and support. Additionally, the large number of enrollees and clickstream interaction logs in any given MOOC provide a vast amount of fine-grained data that can help researchers understand how people learn and how best to support learning in an online environment.

This program of research began with the hope of capitalizing on these properties in order to examine the impact of MOOC discussion forum use on learning outcomes. Simply put, we wanted to study whether viewing discussion board threads while doing homework resulted in final exam gains attributable to this behavior, i.e. controlling for other factors. It seemed prudent to try to account for enrollees with different levels of prior ability and engagement/motivation, as MOOC students are known to have diverse populations. Thus, final exam performance would be our outcome variable; prior ability, engagement/motivation (or some proxy), and discussion forum usage would be covariate predictors. Along the way, however, we perceived that the challenges of operationalizing all of the variables gained more and more importance to the validity of our inferences.

Indeed, recent work by other authors concentrated on the sensitivity of analytical inferences to operationalization of predictor variables such as time-on-task estimation [18]. In reference to that work, this paper may also be seen as an attempt to "penetrate the black box" of a particular MOOC analysis. Thus, we raise the following auxiliary research questions: Does the method of quantifying discussion forum use significantly impact the analysis of its effect on performance? Given that motivation matters, does the decision of which filter to use to exclude unmotivated students change the results of the analysis? Issues of prior ability estimation are myriad; we discuss these briefly below but get into more details in a separate study [4].

In the remainder of this paper, we examine the impact of methodological decisions on the quality and type of inferences that can be drawn from examining MOOC forum use, focusing specifically on methods of quantifying discussion forum use and filtering unmotivated students.

The organization is as follows. By way of motivating our original substantive questions, we first review related literature on the impact of discussion forums in online learning. We then describe our data set. Next, we turn to the challenges of MOOC analyses, in general and specifically to the variables under consideration. We describe different methods for and results from operationalization choices with regards to discussion forum usage, motivation proxies, and prior ability estimates. Finally, we consider the impact of these variables on performance using multiple linear regression models for final exam score.

DISCUSSION FORUMS IN ONLINE LEARNING

The impact of discussion forums on learning in MOOCs and other online courses is still not well understood,

although the literature on the subject dates back to the 1990s. While some early research on discussion forums cautioned about the shortcomings of computer-mediated dialogue as compared with face-to-face interactions [25], much of that research explored the benefits of the cognitive processes involved in the use of discussion forums, such as reshaping ideas and constructing meaning with the help of peers [3,21]. Later research (but still prior to the MOOC era) focused on measuring the level and quality of student activity in the forums, for example using data mining and text mining [8]. Cultivation of successful asynchronous discussion was linked to measures of discussion quality [2]. Artificial intelligence approaches for classifying effective synchronous collaborative learning [23] were also applied to asynchronous forums in a graduate level course [24].

Correlations of discussion activity with external performance measures have been the subject of several studies ranging from high school [15] to college [17,19] to graduate school [24], with mixed results. Correlations of 0.51 were found for topical student discussion behaviors (coded by hand) with concept-test performance in a physics course using the learning online network with computer-assisted personalized approach (LON-CAPA) learning management system [17]. Operationalizing discussion behavior purely by counts, [15] found correlations of 0.27-0.44 between project performance and activity volume in the forums for secondary school computer science. [19] performed a multiple regression analysis of quiz scores in two college psychology courses, finding that only content-page-hits were significant, not counts of discussion posts or reads. [24] also found no significant correlations between number of posts and student success in a graduate level course, but success variability was very low and the number of students was only 18.

Prior to MOOCs, the largest number of students in any of these studies was 214 [17]. This is one profound difference in the MOOC era, where tens of thousands of students participate and often thousands complete an online course. More recent analyses of discussion forum use in large MOOCs include the following: one analysis found that superposters elicited more posting from their less prolific peers, but the study did not analyze the impact of posting behavior on performance [14]. A randomized controlled trial comparing students with access to chat and discussion forms to students with access to only discussion forums found no differences in retention or performance between groups [6]. Background characteristics of forum users and the communication networks they formed were analyzed in [12], which found that higher performing students participated more in discussion forums but did not interact exclusively with other higher performing students.

MOOC DATA SET

The data for this study come from the Spring 2012 Circuits and Electronics MOOC on the MITx platform. Descriptive measures of discussion forum usage, homework performance, and final exam scores were extracted from the MOOC clickstream logs using parsers written in Python [22]. Over 100,000 students registered for this course, though only half as many attempted to solve at least one problem in the course. Roughly 9000 attempted at least one problem on the final exam, and 7157 earned certificates.

Each access by a student to the discussion forum was recorded in the click-stream logs of the MOOC, as were the times when the student first opened each weekly homework assignment and the time of the last submit (the "homework window"). Thus it was straightforward to enumerate the number of threads viewed each week during the homework window. In this course, the most commonly referenced resource during homework solving was the discussion forum [22], which was structured as a Q&A board with up-voting and search capability (other course resources included lecture videos, an online textbook, and a wiki). Interestingly, most of this activity was "voyeuristic" not contributive: 67% of active students viewed (that is, clicked on—without scroll information and/or eye-tracking sensors, one cannot say for sure whether students read the threads they opened) at least one discussion thread between the first time they opened the homework and their last submission, whereas fewer than 10% posted a question, comment, or answer. Moreover 95% of all discussion activity in this course (by number of events) was viewing, not posting.

Because discussion forum content was generated by students, the forum was not as rich in the first few weeks of the course until participation reached a critical level, as shown in Figure 1.

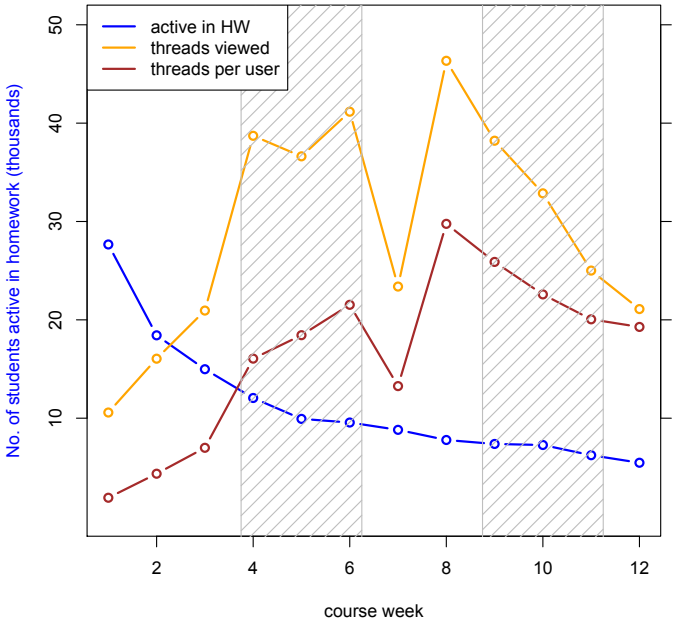


Figure 1: MOOC activity over time. Grey bars indicate early stage and late stage intervals on either side of the midterm.

As seen in this figure, the number of students actively doing homework in our data set (active in homework, blue line)

decays over time, while activity in the forums increases before leveling off (threads per user, brown line). The midterm exam occurred between weeks 7 and 8, which explains the dip and then surge in discussion forum activity, as it was not permitted to post questions or answers about the midterm. The greyed regions of Figure 3 represent two three-week intervals, which we label "early stage"—weeks 4-6, after the discussion forum had fully taken off but before the midterm—and "late stage"—weeks 9-11, after the midterm but before the final exam. To smooth out week-to-week variation, we summed over views within each three-week long interval, as discussed below.

CHALLENGES IN OPERATIONALIZING PREDICTORS

MOOCs differ from standard courses in a number of ways that make analyzing enrollee behavior difficult. These include higher than usual variability in prior educational attainment [20] and assessment motivation [26], large amounts of missing data, and affordances of multiple attempts on both formative and summative assessments [4]. Due to these issues, several researchers have noted that traditional measures of participation and achievement may need to be reconsidered in the context of MOOCs [5,7,13,16]. In this section, we introduce three sets of challenges, one for each predictor variable:

1. How can *prior ability* be estimated so that performance models can control for prior ability?
2. How should *discussion forum usage* be quantified? Is it a static quantity, or does it change over time?
3. Can we identify students who appear to be *disengaged/unmotivated*? What effect would excluding those students have on the effect size of forum usage?

Prior Ability

Enrollees in MOOCs range from high school students to professionals with earned doctorates [20]. Because overall performance is likely to depend on prior ability, this factor should be accounted for in any analysis of "treatment effects" from discussion forum usage. However, prior ability is typically unavailable information. Not all MOOCs survey incoming students, and those that do often survey sparsely. Enrollees in the Spring 2012 Circuits and Electronics MOOC were not given a pretest. Therefore, prior ability had to be inferred from the course data. In this study, we chose to estimate prior ability levels from performance on homework assignments in the first three weeks of the course, when enrollees had just begun to learn the content and before discussion forum use had taken off. The main idea was that early stage ability estimates were not likely to be affected by discussion forum usage, whereas final exam performance might be.

Because homework assignments allowed an unlimited number of attempts, the variability of the eventually correct (EC) score (the official score of record) was quite low. However, scoring items based on whether they were solved correctly on the first attempt (CFA) resulted in a far more normal distribution (see Figure 2). A host of options for scoring homework in the presence of missing data and multiple attempts was described in [4]. While approaches based on polytomous item response models were most predictive of final exam scores, a reasonable improvement of the EC score was obtained for observed scores based on CFA. For simplicity, we use the mean CFA score, which is the proportion of homework problems attempted by each enrollee in the first three weeks of the course that were solved correctly on the first attempt. Skipped items are ignored, rather than scored as incorrect. For detailed considerations of homework scoring in MOOCs, we refer the reader to [4].

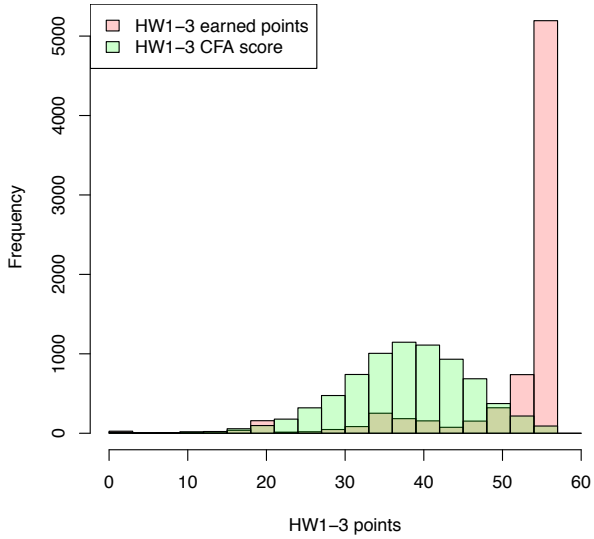


Figure 2: EC and CFA score distributions

It should be noted that the issues of homework scoring also arise in the final exam, which is our outcome measure. We do not consider alternate scoring options, e.g. CFA scoring or item response theory, for the final exam. Only three attempts were allowed versus unlimited attempts on homework, and we did not want to punish students for strategically using their available attempts. However, there remain issues of examinee motivation, as discussed below.

Discussion Forum Usage

The average number of threads viewed per week was shown in Figure 1. We now explore the distribution over MOOC users of the early stage and late stage intervals (grey regions in Figure 1; the purpose of summing was to smooth out week-to-week variation.) We are interested in knowing both the distribution of counts within each interval—e.g. is it simple or bimodal?—as well as across the intervals—i.e. do learners exhibit consistent discussion usage over time, or does it change? These are important considerations for modeling the effect of discussion views. Consider students who purposefully increase their reference to forums after the midterm and reap performance gains as

a result. Modeling their usage as constant over time would distort the positive effect.

As shown in Figure 3, the early/late view count variables are of mixed type: many students do not view any threads, but among those who view at least one, the counts are roughly log-normally distributed. We have added 0.37 to all counts, such that after log-transformation, the students with zero counts appear in the disjoint bin at -1. As seen in the figure, there are roughly 1600 students in this bin for both early stage and late stage counts.

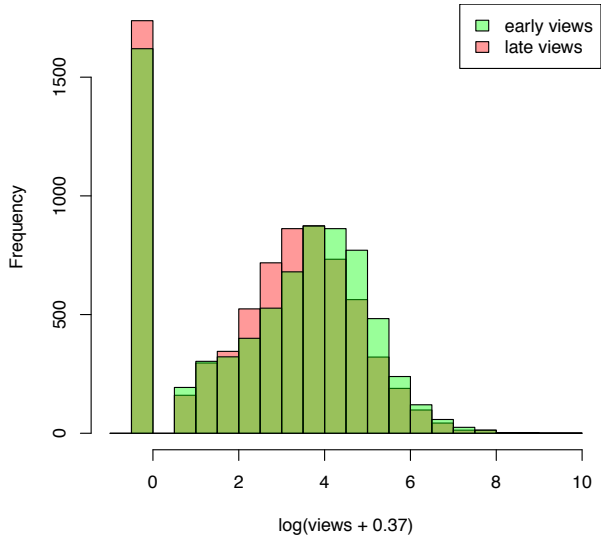


Figure 3: Distribution of view counts (log-transformed)

Figure 3 does not reveal whether there are students who significantly increase or decrease their discussion viewing between these time periods. Moreover, determining what amount of change is significant is a subtle point.

To address this question, we plot early view counts (scaled) against the difference between early and late counts (also both scaled) in Figure 4. Scatterplot and point density are both shown. There is a floor effect, which appears as a diagonal lower bound in the figure, representing students who went from a finite number of threads viewed in the early period to zero in the late period. Another salient feature is that for medium to large values of early counts, the change (from early to late counts) seems to be a random effect around zero (no change). This random description does not however fit all of the data. There does appear to be a clump of students on the upper left, whose viewing counts increase from very low levels to moderate levels. And there are some whose viewing decreases beyond the noise threshold. We chose to identify these students as outliers from the random distribution.

We determined empirical means and variances after removing low values and then drew a random sample of 7000 data points from a bivariate normal distribution with center $\mu = (4.17, -0.27)$ and with covariance matrix $\Sigma = (1.15, 0, 0, 0.84)$. Elliptical contours are drawn at the 95% and 99% confidence level in the figure. We have also included reference lines at the vertical mean value plus and minus log(2). The purpose of this second boundary is to define a criterion for those students whose early view counts were extreme outliers but whose change was still modest. Since the vertical axis is a difference of logarithms (or the log of the ratio), points outside this inner region represent doubling (or halving) in the counts.

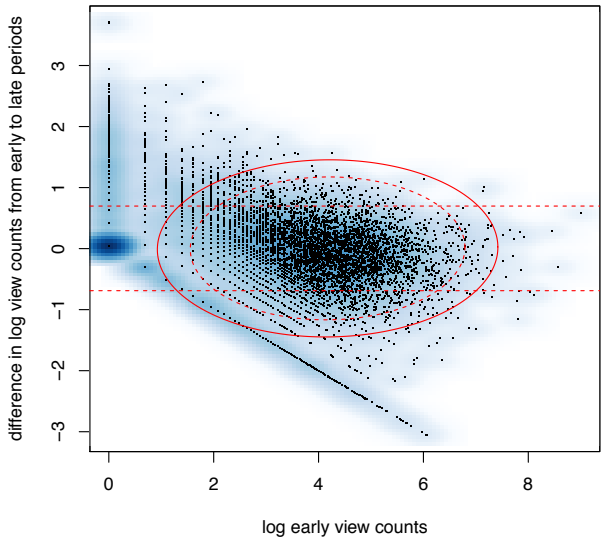


Figure 4: Change in discussion view counts against early counts. Ellipses denote 95% and 99% confidence intervals around a bivariate normal uncorrelated distribution. Dashed lines at +/- log(2) denote doubling thresholds.

As a result of this exploratory analysis, we divided our initial population into an *overall group* (N = 6505), whose discussion viewing during homework could be seen as unchanging over time and thus aggregated into a variable V_O, and a *change group* (N = 989), whose viewing change V_C should be modeled instead. V_O is the sum of the early and late stage counts, and V_C is the difference. Each would subsequently be treated as a continuous variable in an overall model or a change model, respectively.

Assessment-oriented Engagement and Total Time as Proxy Measures of Motivation

Inferences about ability from standard measures of performance may not always be valid in a MOOC due to differences in enrollees' motivations for taking the course. The expectancy-value model [9] puts the validity problem as follows: achievement motivation is influenced by both the individual's expectancies for success and the subjective value attached to success on the task. If the value of success is low, the examinee's achievement motivation will be low. Motivation thus acts as a source of construct-irrelevant variance and impacts the validity of score-based inferences [10]. In a meta-analysis of twelve empirical studies, [26] found that motivated students scored on average 0.59

standard deviations higher than their unmotivated counterparts. Such a result highlights the need to evaluate examinee motivation and possibly filter data from unmotivated test-takers to strengthen the assumption that a score obtained from an assessment accurately reflects the underlying abilities/traits of interest [1].

Consider the final exam score, which typically counts heavily toward qualification for a certificate (in the course under study, the final counted for 40% of the cumulative grade). However, the MOOC certificate is largely symbolic when it confers no degree credit. Thus, enrollees whose motivations for taking the course do not include certification may well view the final exam as low-stakes. The consequentiality of certificates may, in fact, change as more MOOCs seek accrediting status and even charge fees accordingly.

In the following, we consider three solutions to this problem, which is essentially the problem of whom to include. The first is to use a heuristic cutoff with respect to proportion of items attempted in the initial and final ability assessments. In the second solution, we attempt to filter out unmotivated students using a simple measure that should be relatively insensitive to the initial and final assessments, namely total time spent online in the course. The third and most intricate solution will be to use a latent class cluster analysis to model the course population as a mixture of classes based on cumulative evidence of assessment-oriented engagement (AOE). Thus both AOE and time-on-task are effective proxy measures for motivation, but we continue to use the original term in order to make contact with validity literature.

Motivation heuristic filter on attempts

Screening out students who attempted less than 60% of the HW1-3 items (which constitute our proxy measure of "prior ability") or less than 60% of the final exam leaves 6210 students. This proportion is chosen to match the passing grade threshold of the course; in order to achieve this minimum, a student must at the very least attempt the same fraction of assessment items. This cutoff ignores the proportion of attempts on items in between Week 3 and the final exam, which will enter into the latent class analysis.

Although this is a filter based on attempts and not scores, it raises selection bias issues. While low-performing students who at least attempted many items would remain, this filter does, by definition, remove low scoring students. Thus our proxy for motivation is wrapped up in the outcome variable of our analysis. The rationale for solution two is partly a response to the bias of solution one.

Motivation heuristic filter on time

What if there were students who invested significant amounts of time and effort in this course but were simply unable to answer many questions and were disinclined to guess? Alternately, what if there were students who carelessly attempted many items, but whose investment in the course was more accurately reflected in low overall time commitment. Rather than filter on proportion of assessment items, we considered overall time spent in the course as a proxy for motivation. All activity, including video views, was included in this time aggregate, which is roughly log-normally distributed (slightly skewed to the left) with a median value around 100 hours. At a minimum time cutoff of 30 hrs (~1.5 standard deviations below), 679 students would be excluded, leaving 6815.

Motivation via latent class analysis of AOE

In the third approach, rather than determine whom to include or exclude, we seek to identify self-similar groups of students based on a pattern throughout the course. We could then model the effect of discussion viewing separately for all groups. Our idea is related to the approach in [16], where week-by-week trajectories were clustered. The results of that analysis were largely interpreted in terms of proportion of assessment attempted, so we went directly to that measure as a basis for clustering. We used five measures based on proportion of assessment items attempted: homework in weeks 1-3, homework in weeks 4-6, midterm exam, homework in weeks 9-11, and final exam. Each student's record of item attempts was thus mapped to a vector of five proportions, and these vectors were clustered using the Gaussian mixture model-based clustering algorithm in the MClust package [11] in R.

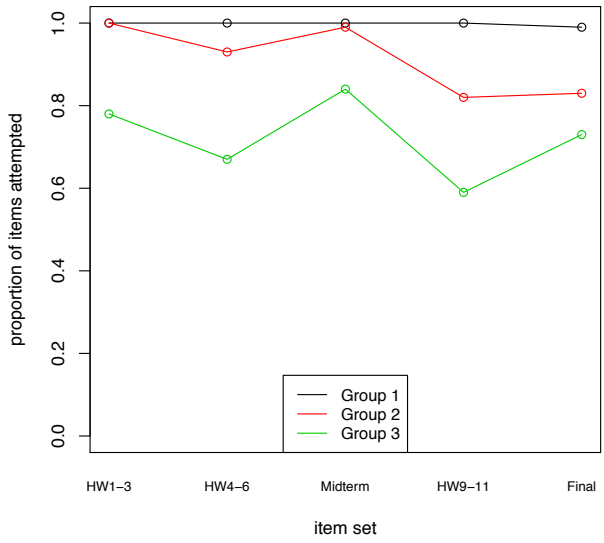


Figure 5: Mean values of proportion of items attempted for three latent class cluster groups.

The model-based approach used here differs from the clustering method in [16], but the results are consistent. The best fit was at three clusters. Mean values for proportion of items attempted are plotted in Figure 5. Groups 1-3 roughly correspond to what [16] called completing, disengaging, and sampling. Probably because we removed in advance students who did not attempt at least one final exam

problem, we do not have an auditor group, typified by students who watch videos but do not attempt any assessment items.

SUBSTANTIVE ANALYSES

Having operationalized our predictors, we now turn to modeling the effect of discussion viewing on final exam performance. Using multiple linear regression, we examine the standardized regression coefficient for the discussion viewing term as a probe of effect size. Based on the exploratory analyses described above, discussion viewing was treated differently for those students whose usage levels were consistent overall versus those who changed their viewing amount between the early and late stages. We computed two different variables V_O and V_C for these two populations respectively. Variability in motivation was handled both through heuristic attempt-based and time-based filters as well as via latent class analysis.

Model and results using motivation filters

Consider the following linear model for predicting the final exam Y using prior ability θ and *overall* discussion view counts V_O,

$$Y = \beta_0 + \beta_1\theta + \beta_2 V_O$$

The change model is identical except for the substitution of view change for overall views. Importantly, the populations included for each model are different, as described above.

Table 1 reports standardized regression coefficients β_2 for these two models. The first column is the result when including all students who attempted at least one final exam problem and one homework item in weeks 1-3 (HW1-3 performance was the basis for estimating prior ability θ). The middle column shows results when excluding students who spent fewer than 30 hours online. The last column shows results excluding those who did not attempt at least 60% of *both* the final exam and the weeks 1-3 homework.

Table 1: Standardized regression coefficients for discussion viewing factor in two models under different data thresholds (white cells p < .001; grey cells not significant)

	No filter	Time > 30h	Attempt > 60%
Overall β_2	0.24	0.18	-0.01
Change β_2	0.19	0.19	0.16

The effect of discussion viewing in the overall model (first row of Table 1) appears to be significant when no filter is applied. But this unfiltered population contains hundreds of students who attempted very few assessment items, so these coefficients are not necessarily trustworthy. Indeed, the effect of overall viewing starts to decline as the population is refined in the next two columns. Screening out students who spent comparatively little time in the course reduces the effect but not by much. On the other hand, after screening out students who did not *attempt* at least 60% of those assessment items that formed the basis of the prior and outcome performance measures, the effect of discussion viewing disappears entirely.

At the least, it must be said that the effect size of discussion viewing in the overall model is sensitive to selection of students. We note that these models altogether explain only about 10% of the variance in the final exam. The midterm exam, for reference, is more predictive ($R^2 = 0.22$).

The effect of discussion views in the change model (second row), in contrast, appears to be more robust under selection for motivated students. At first glance, it is not clear whether increases in viewing are translating into higher scores or decreases in viewing are translating into lower scores. The latter could be consistent with attrition, for example. However, if attrition were the dominant explanation, then the third column coefficient would also be small, since course droppers would have been screened out. Thus the change model coefficients suggest that increasing discussion views are associated with higher final scores. We believe that interpretation of this effect is improved with reference to the latent class models, described next.

Model and results for latent class analysis

Table 2: Standardized regression coefficients for the overall viewing model with latent class cluster groups (white cells, p < .005; grey cells are not statistically significant)

$Y = \beta_0 + \beta_1\theta + \beta_2 V_O + \beta_3 G + \beta_4\theta G + \beta_5 V_O G$							
	0.75	0.14	-0.09	0	0	0	G=1
				-0.76	0.05	0.05	G=2
				-0.96	0.09	0.53	G=3

In Table 2 we show the model equation and estimated parameters for overall viewing effect with latent class cluster assignments. There were significant interactions between the cluster groups G and the continuous prior ability and discussion variables for the overall model; therefore we include five coefficients. Group 1, the reference group, attempted almost all assessment items (see Figure 5). Because Group 2 and 3 attempted fewer items, the main effect for those groups ($\beta_3; p < .001$) is a lower expected final exam score. Indeed, Group 1 may be thought of as a more restrictive subsample from the third column of Table 1. The interpretation of this small negative β_2 is not necessarily that discussion views hurt, of course. Among Group 1 students, more viewing may indicate challenges with homework that transfer into challenges on the final.

Given that students in Group 3 omitted significant numbers of assessment items, why would such students reap more rewards from viewing discussion threads (β_5)? A possible explanation is that discussion viewing is a proxy for activity within Group 3. Indeed, there were positive correlations

between overall views and final exam items *attempted* (0.38) as well as late-stage homework *attempted* (0.53). Students who viewed more also did more assessment items relative to other students in this group.

Finally, Table 3 shows the change model with latent classes. Comparing to the second row of Table 1, we see now that for Group 1, increasing views are no longer associated with higher final exam scores. Recall that this group comprises the most active population with respect to assessment items. Again, a plausible explanation is that increasing discussion views are simply an indication of increasing participation in Groups 2 and 3, for example due to late joiners to the course. The correlation between viewing change and final exam items attempted is low in both cases (roughly 0.06), but the correlation with late homework attempted is moderate (0.27 and 0.33 for Groups 2 and 3, respectively). For the sporadic users of assessment in these groups, the positive association of increasing discussion views over time is there, but it may be linked to increasing engagement with the homework.

Table 3: Change model including latent class cluster groups (white cells, p < .05; grey cells are not statistically significant)

$Y = \beta_0 + \beta_1\theta + \beta_2 V_C + \beta_3 G + \beta_4 \theta G + \beta_5 V_C G$							
	0.76	0.18	-0.05	0	0	0	G=1
				-0.80	-0.06	0.22	G=2
				-1.14	-0.15	0.21	G=3

CONCLUSIONS AND FUTURE WORK

We started out with a simple goal of studying the learning outcome benefit from viewing discussion threads while doing homework in a MOOC. Along the way, it became clear that operationalizing almost all of the variables in this equation presented challenges. We have considered solutions to several issues that are endemic to MOOCs: estimating prior ability; determining whether to use an overall or a change model of discussion viewing; and screening out unmotivated students for the purpose of increasing the validity of inferences.

In the end, neither overall discussion viewing (for those whose viewing was fairly steady) nor change in discussion view volume appeared to be significant for students who attempted most of the assessment items, i.e. Group 1. The gain that appears from a naïve application of a linear model to the larger student sample (Table 1, column 1) seems to be due to confounding discussing thread viewing with participation, among sporadic participants. More work would need to be done to decouple use of the discussion forum from assessment-oriented engagement, for example by treating the latter as a continuous measure rather than as an indicator on which to filter the population. Moreover, counting discussion thread views is a limited window into usage of the forums. We did not analyze posting or commenting in this analysis, nor did we discriminate between threads using textual analysis.

We did not say much about why the effect size of discussion viewing seemed insensitive to filtering students by overall time spent online. We suspect this is because there were hundreds of students who scored very highly on the final exam in this course but spent almost no time learning; in other words, these students already knew the content, but took the tests for fun or for the certificate.

As suggested above, we suspect that late joiners—whose increasing viewing over time appeared to associate with score gains—were a foil in this analysis. It would be interesting to dig deeper into how to model students whose trajectories of participation are increasing or decreasing over time. Also, although we used the final exam because it was an obvious choice, it may be possible to model the effect of discussion viewing on homework performance directly. There are subtleties to this, because multiple attempts increase the likelihood of correct responses. From a learning science perspective, looking at how students search the forums to get homework assistance may also be a fruitful direction.

ACKNOWLEDGMENTS

We are grateful to edX for providing the raw data for this analysis, to Daniel Seaton for critical contributions to the processing of these data, and to helpful suggestions from reviewers. DEP would like to acknowledge support from a Google faculty award and from MIT.

REFERENCES

1. AERA, APA, NCME. *Standards for educational and psychological testing.* American Educational Research Association, Washington, D.C., 1999.
2. Andresen, M. Asynchronous Discussion Forums: Success Factors, Outcomes, Assessments, and Limitations. *Educational Technology & Society 12*, (2009), 249–257.
3. Bates, A.W.T. *Technology, E-Learning and Distance Education.* Routledge, 1995.
4. Bergner, Y., Colvin, K., and Pritchard, D.E. Estimation of Ability from Homework Items When There Are Missing and/or Multiple Attempts. *Proceedings of LAK 2015*, (2015).
5. Clow, D. MOOCs and the funnel of participation. *Proceedings of the Third International Conference on Learning Analytics and Knowledge Discovery*, (2013), 185–189.
6. Coetzee, D. and Hearst, M.A. Chatrooms in MOOCs : All Talk and No Action. (2014), 127–136.
7. DeBoer, J., Ho, A.D., Stump, G.S., and Breslow, L. Changing "Course": Reconceptualizing Educational

Variables for Massive Open Online Courses. *Educational Researcher March*, (2014), 74–84.

8. Dringus, L.P. and Ellis, T. Using data mining as a strategy for assessing asynchronous discussion forums. *Computers & Education 45*, 1 (2005), 141–160.
9. Eccles, J.S. and Wigfield, A. Motivational Beliefs, Values, and Goals. *Annual Review of Psychology 53*, (2002), 109–132.
10. Eklof, H. Development and Validation of Scores From an Instrument Measuring Student Test-Taking Motivation. *Educational and Psychological Measurement 66*, 4 (2006), 643–656.
11. Fraley, C. and Raftery, A.E. Model-based Clustering, Discriminant Analysis and Density Estimation : *Journal of the American Statistical Association 97*, (2002), 611–631.
12. Gillani, N. and Eynon, R. Communication patterns in massively open online courses. *The Internet and Higher Education 23*, (2014), 18–26.
13. Ho, A.D., Reich, J., Nesterko, S.O., et al. HarvardX and MITx: The First Year of Open Online Courses, Fall 2012-Summer 2013. *SSRN Electronic Journal*, (2014).
14. Huang, J., Dasgupta, A., Ghosh, A., Manning, J., and Sanders, M. Superposter behavior in MOOC forums. *Proceedings of the first ACM conference on Learning @ scale conference - L@S '14*, (2014), 117–126.
15. Kay, R.H. Developing a comprehensive metric for assessing discussion board effectiveness. *British Journal of Educational Technology 37*, 5 (2006), 761–783.
16. Kizilcec, R., Piech, C., and Schneider, E. Deconstructing Disengagement : Analyzing Learner Subpopulations in Massive Open Online Courses. *Proceedings of the Third International Conference on Learning Analytics and Knowledge Discovery*, (2013).
17. Kortemeyer, G. Correlations between student discussion behavior, attitudes, and learning. *Physical Review Special Topics - Physics Education Research 3*, 1 (2007), 010101.
18. Kovanović, V., Gašević, D., Dawson, S., Joksimović, S., Baker, R.S., and Hatala, M. Penetrating the black box of time-on-task estimation. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge - LAK '15*, ACM Press (2015), 184–193.
19. Ramos, C. and Yudko, E. "Hits" (not "Discussion Posts") predict student success in online courses: A double cross-validation study. *Computers & Education 50*, 4 (2008), 1174–1182.
20. Rayyan, S., Seaton, D.T., Belcher, J., Pritchard, D.E., and Chuang, I. Participation And performance In 8.02x Electricity And Magnetism: The First Physics MOOC From MITx. (2013), 4.
21. Rowntree, D. Teaching and learning online: a correspondence education for the 21st century? *British Journal of Educational Technology 26*, 3 (1995), 205–215.
22. Seaton, D.T., Bergner, Y., Chuang, I., Mitros, P., and Pritchard, D.E. Who does what in a massive open online course? *Communications of the ACM 57*, 4 (2014), 58–65.
23. Soller, A. Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in ... 12*, 1 (2001).
24. Song, L. and McNary, S. Understanding students' online interaction: Analysis of discussion board postings. *Journal of Interactive Online Learning 10*, 1 (2011), 1–14.
25. Thomas, M.J.W. Learning within incoherent structures: the space of online discussion forums. *Journal of Computer Assisted Learning 18*, 3 (2002), 351–366.
26. Wise, S.L. and DeMars, C.E. Low Examinee Effort in Low-Stakes Assessment: Problems and Potential Solutions. *Educational Assessment 10*, 1 (2005), 1–17.

Influence Analysis by Heterogeneous Network in MOOC Forums: What can We Discover?

Zhuoxuan Jiang[1], Yan Zhang[2], Chi Liu[1], Xiaoming Li[1]
Institute of Network Computing and Information System
Peking University, Beijing, China
[1]{jzhx,liuchi,lxm}@pku.edu.cn, [2]zhy@cis.pku.edu.cn

ABSTRACT

With the development of Massive Open Online Courses (MOOC) in recent years, discussion forums there have become one of the most important components for both students and instructors to widely exchange ideas. And actually MOOC forums play the role of social learning media for knowledge propagation. In order to further understand the emerging learning settings, we explore the social relationship there by modeling the forum as a heterogeneous network with theories of social network analysis. We discover a specific group of students, named representative students, who feature large engagement in discussions and large aggregation of the majority of the whole forum participation, except the large learning behavior or the best performance. Based on these discoveries, to answer representative students' threads preferentially could not only save time for instructors to choose target posts from all, but also could propagate the knowledge as widespread as possible. Furthermore if extra attention is paid to representative students in the sight of their behavior, performance and posts, instructors could readily get feedback of the teaching quality, realize the major concerns in forums, and then make measures to improve the teaching program. We also develop a real-time and effective visualization tool to help instructors achieve these.

Keywords

MOOC forum, Coursera, influence, behavior, performance, heterogeneous network

1. INTRODUCTION

Comparing with the traditional distance education or online courses, discussion forums in Massive Open Online Courses (MOOC) offer a big and lively venue for communication between students and instructors, which has been proved important for large-scale social learning [1, 7, 9]. However, due to their massiveness, the forums are full of various information relevant and irrelevant to the course [6]. So how to fast and accurately extract valuable information from the large-scale settings has become a problem to which priority should be given.

Considering Twitter, Facebook or StackOverflow, MOOC forums look similar to some kind of social media because of the large number of participants and their interactivity. Every member in the forum may talk about course content, such as asking or answering a question. The intensive interaction between them actually supports the knowledge propagation between members of the learning community. However here comes up a dilemma. In light of knowledge propagation, the proportion of instructors' responses is expected as large as possible in order to resolve students' questions; But considering the scale, instructors could not have enough time to read every thread. In order to cope with this situation, we propose a trade-off solution that extracts influential students from all and recommended them to instructors. Then instructors could make decisions in a much smaller scale and their' effort would be amplified based on principles of influence propagation [12, 16, 24].

Although the definition of influence is various from different perspectives, we leave aside others except instructor for the time being in this paper. We conceive in each forum there could be a group of influential students who attract many others to interact with them, just like the verified accounts in Twitter. We call them 'representative students' and they involuntarily undertake the responsibility for knowledge propagation. So instructors could amplify the influence of right answers by preferentially responding to questions of representative students. Thus, many more students who pay attention to representative students' answers would also benefit without actually having a response by the instructor. On the other hand, given that representative students' threads may get a lot of attention, instructors could address the main concerns in the learning community more promptly. Through the rank list of representative students' influence, the chief instructor could also realize whether other instructors (or called TAs) are on duty, since TAs' influence could be calculated meanwhile. As we show later in this paper, representative students' performance is not the best within the learning community, but given their positive motivation and high volume of messages answering promptly their questions is beneficial for the whole learning community.

Since posts irrelevant to the course are unavoidable in such a free forum, for example chatting, making friends or other things, it is not reasonable to directly regard superposter [9] as representative students or merely consider their social relationship. Experiments later in this paper approve the opinion and find post contents are useful. That being the case, since we regard the interaction in MOOC forums as the procedure of knowledge propagation in social media, we could build a heterogeneous network [23] to model the forum with two kinds of entities by leveraging theories of networked entities ranking. Then we can get a rank list of students' in-

fluence from that network with a specially designed algorithm. The higher a student ranks on the list, the more influential she would be. This model could fully utilize the social information and textual messages to avoid outliers or exceptions (e.g. someone who always submits posts irrelevant to the course).

To our knowledge, this is the first work to adopt a heterogeneous network to model social relationship in MOOC forums and extract representative students. We also propose a novel algorithm for ranking students' influence based on graphic theories. Experimental results show the effectiveness and efficiency of the algorithm are both decent. Through the analysis of representative students' log data, we find they engage highly and aggregate much participation except the excellent grades, which suggests they are representative for instructors to watch the class and are the first low hanging fruit for increasing the passing rate. Analysis of historical records of interaction between instructors and students indicates it is time-saving and meaningful for instructors to recommend threads of representative students. Based on those discoveries, we developed a web service of visualization tool as an assistant for instructors to achieve the conception of supervising their class effort-savingly.

2. RELATED WORK

In traditional off-line classes, the scale is relatively small and face-to-face Q&A is not a challenge. And in traditional online education or online video class, not only the scale is not large enough but the absence of instructors is very common. However, a widespread viewpoint is that it is quite important for MOOC to make students engage in a social learning environment to guarantee and improve the teaching quality [1, 6, 7, 18].

In view of researches in the field of Community Question Answering (CQA), issues related to this paper are about expert finding and forum search [21]. Recently, several novel methods for finding experts in CQA have been provided [26, 29, 30]. Nevertheless, there would be rare experts in MOOC forum due to the specificity that a MOOC forum is not open to all kinds of discussions and it just belongs to the corresponding course for students to acquire knowledge. Also the definition of representative students here is different from that of experts. On the other hand, the task of discovering representative learners and their posts seems like forum search [3, 19] which develops a mechanism analogous to a search engine. But here we concentrate on just the ranking result and not emphasise the accuracy of retrieval. Except those general forum-related work, recently some researches of MOOC forums have been published from various perspectives. For example, Yang et al. [25] tried thread recommendation for MOOC students with method of an adaptive feature-based matrix factorization framework. Wen et al. [22] analyzed the sentiment in MOOC forums via students' words for monitoring their trending opinions. And Stump et al. [20] proposed a framework to classify forum posts.

The classical PageRank [5] and HITS [14] have been applied on broad problems of networked entities ranking and been promoted to solve problems in heterogeneous network [11, 15, 27]. [17, 28] built a heterogeneous network with two types of nodes to discover the influential authors with scientific repository data, which is similar to our work. The point in common is to discover influential entities with iteration by building a graphic model. In this paper, we leverage that principle and build a new heterogeneous network to model MOOC forum and discover representative students.

Besides, many MOOC log analysis also involve forums. Anderson et al. [2] deployed a system of badges to produce incentives for activity and contribution in the forum based on behavior patterns. Huang et al. [9] specially analyzed the behavior of superposter in 44 MOOC forums and found MOOC forums are mostly healthy. Kizilcec et al. [13] did a research on the behavior of students disengagement. Some technical reports and study case papers also involved behavior analysis of MOOC students in forums, such as [8] and [4]. Nevertheless, we believe incentives established on intelligent analysis of various data like social information and textual messages would be more reasonable than on the pure credits mechanism in traditional forums, since the latter only considers the quantity of behavior while not the quality.

Table 1: Pairs of course code and course title

Course Code	Course Title
peopleandnetworks-001	Networks and Crowds
arthistory-001	Art History
dsalgo-001	Data Structures and Algorithms A
pkuic-001	Introduction to Computing
aoo-001	The Advanced Object-Oriented Technology
bdsalgo-001	Data Structures and Algorithms B
criminallaw-001	Criminal Law
pkupop-001	Practice on Programming
chemistry-001	General Chemistry (Session 1)
chemistry-002	General Chemistry (Session 2)
pkubioinfo-001	Bioinformatics: Introduction and Methods (Session 1)
pkubioinfo-002	Bioinformatics: Introduction and Methods (Session 2)

Table 2: Statistics per course

Course	# threads	# posts	# votes
peopleandnetworks-001	219	1,206	304
arthistory-001	273	2,181	1,541
dsalgo-001	283	1,221	266
pkuic-001	1,029	5,942	595
aoo-001	97	515	204
bdsalgo-001	319	1,299	132
criminallaw-001	118	763	648
pkupop-001	1,085	6,443	977
chemistry-001	110	591	65
chemistry-002	167	715	678
pkubioinfo-001	361	2,139	1,474
pkubioinfo-002	170	942	235
Overall	4,259	24,042	-

3. DATASET

We use all the log data of 12 courses from Coursera platform. They were offered in Fall Semester of 2013 and Spring Semester of 2014. There are totally over 4,000 threads and over 24,000 posts. For convenience later in the paper, Table 1 lists the pairs of course code and course title. Table 2 shows the statistics of the dataset per course. Here posts denotes responses including posts and comments. We can see both the subjects and scales range widely.

4. MODEL AND ALGORITHM

In order to model MOOC forums as social media, the first challenge is that no explicit post-reply relationship which describes who replies who is recorded. We simplify this problem and assume

Table 3: Attributes of the heterogeneous network constructed per course

Course	G_S			G_K			G_{SK}	
	n_S	$\|E_S\|$	$\|E_S\|/n_S^2$	n_K	$\|E_K\|$	$\|E_K\|/n_K^2$	$\|E_{SK}\|$	$\|E_{SK}\|/(n_S+n_K)^2$
peopleandnetworks-001	321	3,287	0.032	1,193	104,821	0.074	4,814	0.002
arthistory-001	540	17,022	0.058	3,376	1,019,289	0.089	14,195	0.001
dsalgo-001	295	1,876	0.022	1,152	124,118	0.094	5,009	0.002
pkuic-001	768	19,801	0.034	2,302	302,989	0.057	14,599	0.002
aoo-001	175	1,963	0.064	783	73,208	0.119	2,597	0.003
bdsalgo-001	225	2,369	0.047	781	23,540	0.039	3,133	0.003
criminallaw-001	219	2,971	0.062	1,224	123,737	0.083	4,577	0.002
pkupop-001	628	12,883	0.033	1,748	88,035	0.029	13,807	0.002
chemistry-001	130	886	0.052	1,055	111,026	0.100	2,685	0.002
chemistry-002	125	2,341	0.150	964	61,425	0.066	2,574	0.002
pkubioinfo-001	594	22,275	0.063	686	46,768	0.099	1,946	0.001
pkubioinfo-002	189	1746	0.049	380	16662	0.115	784	0.002

Table 4: Notations

Notation	Description
$G = (V, E, W)$	heterogenous network
$G_S = (V_S, E_S, W_S)$	student subnetwork
$G_K = (V_K, E_K, W_K)$	keyword subnetwork
$G_{SK} = (V_{SK}, E_{SK}, W_{SK})$	bipartite subnetwork
n_S, n_K	$\|V_S\|$, $\|V_K\|$

if two students appear in the same thread, they have the same topic interests and the one whose post is chronologically later replies the other. As mentioned in previous sections, post contents of representative students should be course-related. Thus it may be not enough to cover that demand with only extracting the post-reply relationship. Based on the fact that the most post contents are course-related [9], we add the keywords as another kind of entities into the model to construct the heterogenous network. The keywords here are all meaningful nouns in post contents and they could represent various aspects of topics. Other kinds of parts of speech are unexplored at the present. The role of keywords in the heterogenous network is to help the algorithm reinforce the influence of students who involve more topics, which ensures the need that posts of presentative students are course-related. Figure 1 shows the demo of the heterogeneous network, and Table 4 lists the defined notations.

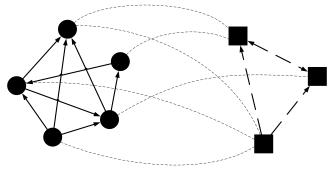


Figure 1: Demo of the heterogeneous network G. Circles denote V_S and rectangles denote V_K. Solid lines with arrows denote the co-presence relationship between students in the same thread and arrows denote one whose post is later points to the other.Dash lines with arrows denote the co-presence of keywords in the same thread but directed or bidirectional arrows mean the two keywords are in the different post or not. Dash lines without arrows denote the authorship between students and keywords. The weight values mean the times of co-presence of two entities on corresponding edges. Self co-presence is meaningless and all ignored.

This model captures the characteristic that representative students would own more latent post-reply relationship and involve more topics. After building the network through log dataset, the basic attributes of graphs per course are calculated (Table 3).

For co-ranking students and keywords, we need an algorithm. We simulates two random surfers jumping and walking in the heterogeneous network and design the algorithm named Jump-Random-Walk (JRW). We assume the weights W represent the influence between entities and the algorithm's task is to discover the most influential students, namely representative students. Figure 2 shows the framework of JRW algorithm.

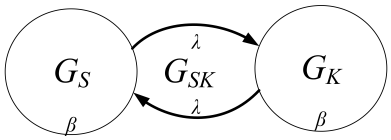


Figure 2: The framework of Jump-Random-Walk algorithm. β is the probability of walking along an edge within G_S or G_K. λ is the probability for jumping from G_S to G_K or in reverse. $\lambda = 0$ means to discover representative students only by using post-reply relationship. We assume the probabilities of each jump or walk are consistent.

Denote $\mathbf{s} \in \mathbb{R}^{n_S}$ and $\mathbf{k} \in \mathbb{R}^{n_K}$ are the ranking result vectors, also probability distributions, whose entries are corresponding to entities of V_S and V_K, subject to $\|\mathbf{s}\|_1 \leq 1$ and $\|\mathbf{k}\|_1 \leq 1$. Denote the four transition matrixes, G_S, G_K, G_{SK} and G_{KS}, for iteration as $S \in \mathbb{R}^{n_S \times n_S}$, $K \in \mathbb{R}^{n_K \times n_K}$, $SK \in \mathbb{R}^{n_{SK} \times n_{SK}}$, and $KS \in \mathbb{R}^{n_K \times n_S}$ respectively. Adding the probability of random jumping for avoiding trapped in connected subgraph or set of no-out-degree entities, the iteration functions are

$$\mathbf{s} = (1-\lambda)(\beta S\tilde{\mathbf{s}} + (1-\beta)\mathbf{e}_{n_S}/n_S) + \lambda SK\tilde{\mathbf{k}}, \quad (1)$$

$$\mathbf{k} = (1-\lambda)(\beta K\tilde{\mathbf{k}} + (1-\beta)\mathbf{e}_{n_K}/n_K) + \lambda KS\tilde{\mathbf{s}}, \quad (2)$$

where $\mathbf{e}_{n_S} \in \mathbb{R}^{n_S}$ and $\mathbf{e}_{n_K} \in \mathbb{R}^{n_K}$ are the vectors whose all entries are 1. The mathematical forms of four transition matrixes are

$$S_{i,j} = \frac{w^S_{i,j}}{\sum_i w^S_{i,j}} \quad where \sum_i w^S_{i,j} \neq 0, \quad (3)$$

$$K_{i,j} = \frac{w^K_{i,j}}{\sum_i w^K_{i,j}} \quad where \sum_i w^K_{i,j} \neq 0, \quad (4)$$

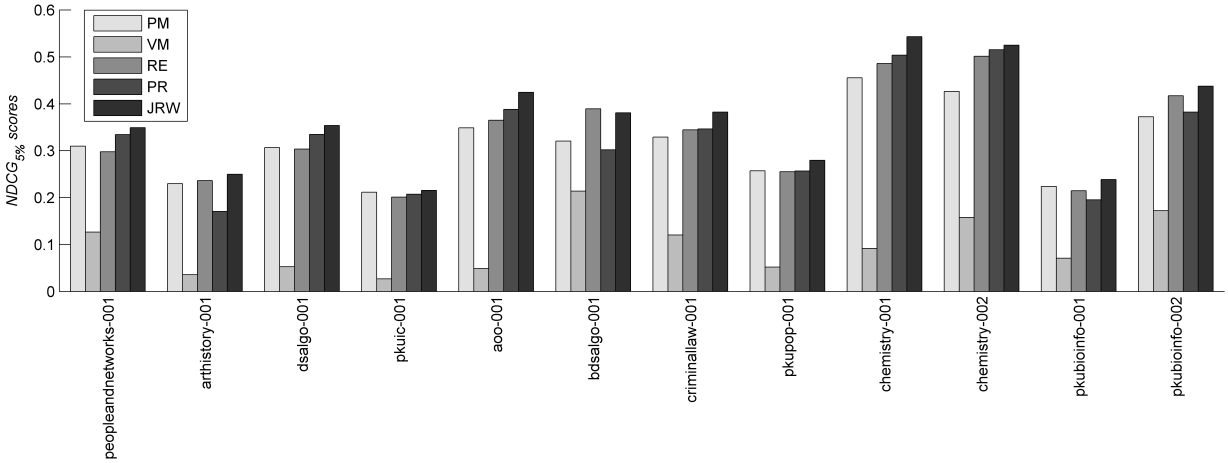


Figure 3: $NDCG_{5\%}$ scores of different rankings

$$SK_{i,j} = \frac{w_{i,j}^{SK}}{\sum_i w_{i,j}^{SK}}, \tag{5}$$

$$KS_{i,j} = \frac{w_{i,j}^{KS}}{\sum_i w_{i,j}^{KS}} \quad where \sum_i w_{i,j}^{KS} \neq 0. \tag{6}$$

$w_{i,j}^S$ is the weight of the edge from V_i^S to V_j^S, $w_{i,j}^K$ is the weight of the edge between V_i^K and V_j^K, $w_{i,j}^{SK}$ is the weight of the edge between V_i^S and V_j^K and $w_{i,j}^{KS}$ is the weight of the edge between V_i^K and V_j^S. Actually $w_{i,j}^{SK} = w_{j,i}^{KS}$. When $\sum_i w_{i,j}^S = 0$, it means the student V_j^S is always the last one in a thread. If $\sum_i w_{i,j}^K = 0$, it means the keyword V_j^K always has no peer in a thread. Actually this situation almost never happens in our filtered data. $\sum_i w_{i,j}^{SK} = 0$ is also impossible, which means every keyword would have at least one author (student). On the contrary, it does not make sure that every student would post at least one keyword, because maybe there is some post having nothing valuable or not containing any nounal keyword. Algorithm 1 shows the detail of JRW algorithm below.

Algorithm 1 Jump-Random-Walk on G

INPUT $S, K, SK, KS, \beta, \lambda, \epsilon$
$\mathbf{s} \leftarrow \mathbf{e}/n_S$
$\mathbf{k} \leftarrow \mathbf{e}/n_K$
repeat
  $\tilde{\mathbf{s}} \leftarrow \mathbf{s}$
  $\tilde{\mathbf{k}} \leftarrow \mathbf{k}$
  $\mathbf{s} = (1-\lambda)(\beta S\tilde{\mathbf{s}} + (1-\beta)\mathbf{e}_{n_S}/n_S) + \lambda SK\tilde{\mathbf{k}}$
  $\mathbf{k} = (1-\lambda)(\beta K\tilde{\mathbf{k}} + (1-\beta)\mathbf{e}_{n_K}/n_K) + \lambda KS\tilde{\mathbf{s}}$
until $|\mathbf{s} - \tilde{\mathbf{s}}| \leq \epsilon$
return $\mathbf{s}, \mathbf{k}$

5. EXPERIMENTS

We do not exclude the data of instructors (or TAs) and regard everyone in the forums as 'students'. So that instructors' influence can also be evaluated in the uniform framework. Since the courses are all in Chinese and the contents are overwhelmingly most in simple

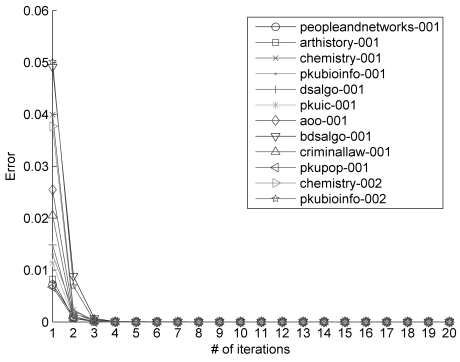


Figure 4: Iteration speed of Jump-Random-Walk

Chinese or traditional Chinese, we filter the non-Chinese contents in the preprocessing step with a tool of Chinese words segmentation which is essential for extracting Chinese keywords. Also we filter the HTML tags irregularly existed. During this process, most spam and valueless posts are filtered incidentally.

To evaluate the effectiveness of JRW, we set some competitors listed below.

- **Post the most** (PM), for superposter by quantity. The more amount and frequency of posts are submitted, the higher she would rank.
- **Be voted the most** (VM), for superposter by quality. The larger ratio of the number of votes earned to the average number of votes in a forum, the higher she would rank.
- **Reputation** (RE), for superposter by reputation. It is a reputation score maintained by the Coursera platform and can be seen as a measure of both the quantity and quality of a forum student's contribution.
- **PageRank** (PR), for representative students only by post-reply relationship. It computes each forum student's influence only in G_S with PageRank algorithm.

Table 5: Representative students' behavior and performance. $P(R|T)$ is the proportion of the number of threads initiated by representative students to the all. $P(R|P)$ is the proportion of the number of posts by representative students to the all. *Over Rate* is the deviation of the average numbers of posts per thread initiated by representative students and the all. $P(R|V)$ is the proportion of the number of watching video by representative students to the all. $P(R|Q)$ is the proportion of the number of submitting quiz by representative students to the all. $P(R|C)$ and $P(R|C,D)$ are the proportions of certificated representative students and certificated representative students with distinction to the all. *Precise* is the proportion of the number of posts by instructors in threads initiated by representative students to that of all the instructors' posts. *Recall* is the proportion of the number of threads replied by instructors to that of threads initiated by representative students.

Course	Forum Behavior			Learning Behavior		Performance		Instructor	
	$P(R\|T)$	$P(R\|P)$	*Over Rate*	$P(R\|V)$	$P(R\|Q)$	$P(R\|C)$	$P(R\|C,D)$	*Precise*	*Recall*
peopleandnetworks-001	0.205	0.246	1.182	0.084	0.074	0.126	0.167	0.267	0.556
arthistory-001	0.289	0.335	1.125	0.102	0.074	0.109	0.188	0.453	0.190
dsalgo-001	0.177	0.355	5.961	0.061	0.082	0.075	0.038	0.182	0.540
pkuic-001	0.282	0.444	-0.649	0.077	0.088	0.117	0.151	0.328	0.545
aoo-001	0.247	0.328	1.446	0.090	0.056	0.071	0.042	0.351	0.583
bdsalgo-001	0.210	0.473	0.401	0.110	0.047	0.047	0.054	0.286	0.866
criminallaw-001	0.246	0.326	1.524	0.060	0.067	-	-	0.504	0.793
pkupop-001	0.283	0.428	1.122	0.095	0.091	0.126	0.212	0.356	0.596
chemistry-001	0.082	0.367	1.706	0.050	0.076	0.078	0.079	0.207	1.000
chemistry-002	0.413	0.494	0.707	0.056	0.042	0.071	0.036	0.362	0.696
pkubioinfo-001	0.260	0.332	-0.963	0.097	0.061	0.075	0.061	0.284	0.713
pkubioinfo-002	0.200	0.445	0.282	0.029	0.035	0.028	0.035	0.210	0.706

- **Jump-Random-Walk** (JRW), for representative students. It co-ranks the influence of both forum students and keywords meanwhile in G.

In order to compare with superposter, we set the same metric that a student is called a representative student when she is within top 5% of the rank list. Note that other alternative metrics, such as the threshold of an absolute number, are also feasible. The parameters used in JRW are $\beta = 0.85$, $\lambda = 0.5$ and $\epsilon = 10^{-6}$. $\lambda = 0.2$ and $\lambda = 0.8$ are also tried, however the differences are tiny. We adopt Normalized Discounted Cumulated Gain (NDCG) [10] as the metric which is applicable for evaluating rankings' quality. We invited two human judges who both are experienced in MOOC forums. They give the influence of each top 5% student a score by reading all the contents of related threads. Each thread and post here are preprocessed to be anonymous and unordered. Score values include 0, 1, 2 and 3, which denotes strongly disagree, disagree, agree and strongly agree. Finally the two assessments are averaged.

Figure 3 shows the results of human assessment. JRW outperforms others among the majority of courses as well as PR, which suggests the necessity of building such a heterogeneous network for discovering representative students. If instructors would set a rule to incentivize representative students, JRW could also be more objective and fairer than simple rankings based on the quantity of behavior. Here is a phenomenon that students voted the most are not representative. This is maybe by reason that the majority of forum students are actually not used to voting the influential posts while unusual comments earn many. In addition, we carry out the convergence analysis of JRW algorithm. Figure 4 shows this algorithm can converge rapidly and satisfy the requirement of real-time computation in large-scale applications.

6. ANALYSIS OF REPRESENTATIVE STUDENTS

In this section, we would explore the characteristics of representative students in two aspects of behavior and performance. Then based on the model and algorithm proposed, we developed a web service which can help instructors supervise not only the behavior and performance of each student, but also their relative position compared with the average level of the whole class. This service could be competent for instructors to gain feedback of the teaching quality.

6.1 Behavior and Performance

Firstly, we analyze the difference of behaviors between representative and non-representative students from a statistic view. Table 5 shows the proportions of various behavior of representative students to the whole forum students per course. The column of Forum Behavior contains three indicators, among which $P(R|T)$ and $P(R|P)$ reflect the degree of representative students' participation in forums. *Over Rate* indicates if the value is over zero, it means representative students' threads are more popular than the average, and vice versa. The values of the three indicators suggest in most forums representative students' participation is relatively high considering their low ratio, only 5%, and their threads are more popular. In other words, the result here manifests threads of representative students initiate the majority of discussions, not counting in the possible sub-discussions initiated by them within a thread.

The column of Learning Behavior shows the behavior of watching video and submitting quiz by representative students. The values of the two indicators, $P(R|V)$ and $P(R|Q)$, suggest the degree of learning behavior of representative students is relatively low compared with their participation, but still larger than 5%. So we can infer that representative students' learning behavior is just above the average. This also suggests their motivation is positive by judging from the value of $P(R|Q)$ which is related to the final certificate.

The column of Instructor demonstrates the necessity of preferentially answering the threads of representative students. *Precise* suggests instructors spent almost one third energy on answering representative students' questions, while *Recall* suggests instructors have answered about two third, up to overall, threads initiated by

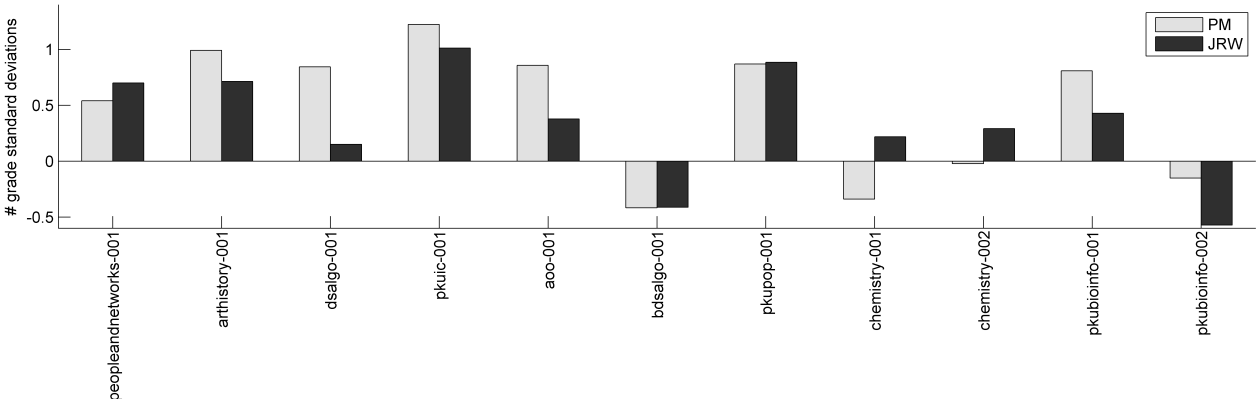


Figure 6: # of standard deviations of representative students outperforming non-representative students on grades per course, comparing with superposters by quantity.

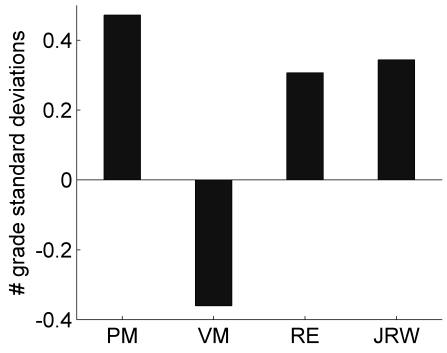


Figure 5: # of standard deviations of representative students outperforming non-representative students on grades averaged over all courses.

representative students. The historical records explain it is necessary for instructors to discover the representative students and their posts, since the range and time cost of choosing which post to reply from all are both reduced. The indicator of *Over Rate* also implies preferentially answering the threads of representative students means more audience would be indirectly beneficial, without actually having a response by the instructor.

Then we would analyze the performance of representative students in the forums. Still in Table 5, the column of Performance denotes the proportions of certificated representative students. $P(R|C)$ and $P(R|C,D)$ are indicators of the passed and the excellent representative students respectively. The values indicate representative students have the higher proportion among the excellent students than the passed students in most courses. However it is potential to improve the proportion of passing rate considering the large forum participation and positive motivation of representative students. So they are worthy being paid extra attention by instructors.

Figure 5 shows the standard deviations, that are averaged z-score grades, to illustrate whether representative students' averaged grade outperforms that of non-representative students among all courses, comparing four different ranking metrics. Superposter by quantity (PM), superposter by reputation (RE) and representative students by JRW (JRW) outperform their peers. However, the score of JRW is lower than that of PM. This may suggest representative students' performance is better than the peers, but not the group with best scores, and the top 5% students who post the most have the higher average score.

From the perspective of each course, representative students' performances are various. Figure 6 exhibits the same standard deviations per course. We can see representative students do not outperform their peers in some courses. Superposter and representative students almost show the consistent trends except for General Chemistry. Representative students' grade is lower than that of superposters by quantity in most courses, which also suggests representative students may have better performance above the average but not the best. This phenomenon could be explained that maybe similar to off-line class, representative students hard to master course content would involve more questions and need more instructions, while superposters by quantity are ones good at the course and always answer questions. So representative students are characterised by large participation of discussions, moderate learning behavior, and above-average performance but not the best.

6.2 Visualization Tool for Instructor

With the various forms of data, an open-and-shut visualization tool could be helpful for instructor to evaluate representative students and supervise their behavior. In order to apply the model proposed in previous sections to an actual function, we scale the final ranking scores to 0-100 as an index score, and developed such a web service whose interface looks as Figure 7.

Here we present the typical usage scenario of the service. Instructors could choose which course to see (Figure 7 A). Surely we would add role and permission administration to protect privacy in the future while here is just the demo of use cases. Then instructors could choose to see how many top students, at most overall (Figure 7 B). Instructors can also select to see the representative students' behavior (Figure 7 C) or their post contents (Figure 7 D). In the main exhibition area (Figure 7 C) where is a table list, instructors can realize the top students' various behavior, including forum participation, learning behavior and performance, students' influence index, and role in the forum. If instructors select to see 'influential post', the main area would replaced by the post contents composed by representative students (Figure 7 D). We conceive that Figure 7 D should provide functions for instructors to re-

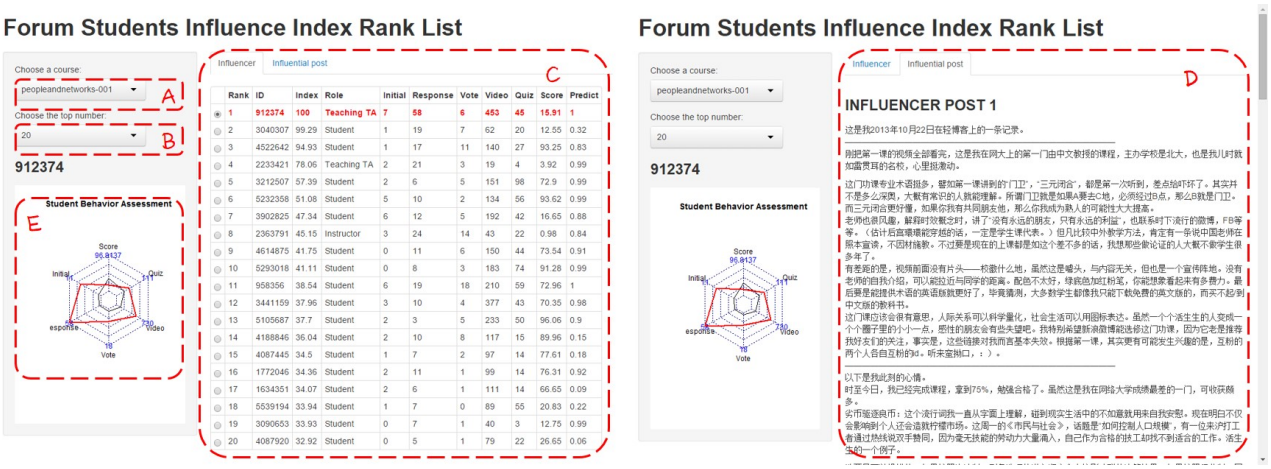


Figure 7: Web service interface

spond, rate, provide feedback and/or other post-related operations like those in the normal forum discussion settings in the future. Given the menu tab 'Influencer' selected, if instructors click the radio button ahead each record of the list, the behavior of corresponding student would also be presented in the radar chart (Figure 7 E). The radar chart displays six dimensions about students' behavior, that are quiz submission, video watching, vote, response, initiated thread, and final score. The scale of each dimension ranges from the minimum to the maximum of each class. Actually there are two closed hexagons on the radar chart. The fixed one in the middle denotes the average values in the whole class while the other, changed with trigger of radio click corresponding to each student, indicates the behavior of individual student. This radar chart can help instructors evaluate the behavior of each student comparing with the whole class under different dimensions.

In our observation and interview, this web service offers instructors the way to realize the class macroscopically and get feedback of main concerns in the forum promptly. Note that due to the rapid speed of our algorithm, this web service can real-timely refresh with changes of students' forum behavior.

7. CONCLUSION AND FUTURE WORK

In the MOOC forum settings, different participants may consider the influence as different definitions. We stand at the side of instructors and assume the influencers in MOOC forums are representative students who stimulate and attract much forum participation. They are actually characterized by lively engagement in forum discussions but unexpected learning behavior and performance, comparing with superposter. They are worthy being paid extra attention from instructors thereby to improve the course passing rate. Since they aggregate much discussion, they could be helpful to amplify instructors' answers and play the latent roles of knowledge propagation. Through representative students' influence, instructors can time-savingly realize the hot topics concerned by the most students. TAs' workload can be evaluated incidentally. In general, it is meaningful for instructors to preferentially read and answer representative students' threads.

In this paper, we leverage methods and algorithms of social network analysis to model MOOC forums in order to further understand the MOOC social learning settings and provide bases for instructors to intervene the social learning. This model has the advantages of fully utilizing social information and textual messages to identify and rank students' influence. Thus based on their behavior, performance and post contents, instructors may make measures to improve the teaching quality, better with that web service of visualization tool as an assistant.

Nevertheless, we have much future work to refine the discoveries in this paper. We would attempt other kinds of heterogeneous networks with more forum information and explore the effect of parameters. Some other random walk algorithms, such as HITS and topic based ones, would be more effective. Furthermore, by integrating our visualization tool into a practical platform, whether the amplification of knowledge propagation via representative students is effective and whether the teaching quality could be promoted still need to be verified through subsequent courses specifically designed in the future.

8. ACKNOWLEDGMENTS

This research was supported in part by 973 Program with Grants No.2014CB340405, NSFC with Grants No.61272340, No.61472013 and No.61370054.

9. REFERENCES

[1] P. Adamopoulos. What makes a great mooc? an interdisciplinary analysis of student retention in online courses. In *Proceedings of the 34th International Conference on Information Systems, ICIS '14*, 2014.

[2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pages 687–698. ACM Press, 2014.

[3] S. Bhatia and P. Mitra. Adopting inference networks for online thread retrieval. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI '10*, pages 1300–1305. AAAI Press, 2010.

[4] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8(1):13–25, 2013.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th*

International Conference on World Wide Web, WWW '1998, pages 107–117. Elsevier Science Publishers, 1998.

[6] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong. Learning about social learning in MOOCs: From statistical analysis to generative model. *IEEE Transactions on Learning Technologies*, 7(4):346–359, 2014.

[7] W. Cade, N. Dowell, A. Graesser, Y. Tausczik, and J. Pennebaker. Modeling student socioaffective responses to group interactions in a collaborative online chat environment. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 399–400. Chapman & Hall/CRC Press, 2014.

[8] HarvardX and MITx: The first year of open online courses, Fall 2012-Summer 2013. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2381263`.

[9] J. Huang, A. Dasgupta, A. Ghosh, J. Manning, and M. Sanders. Superposter behavior in mooc forums. In *Proceedings of the first ACM Conference on Learning @ Scale Conference, L@S '14*, pages 117–126. ACM Press, 2014.

[10] K. Jarvelin and J. Kekalainen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, pages 41–48. ACM Press, 2000.

[11] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 1298–1306. ACM Press, 2011.

[12] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pages 137–146. ACM Press, 2003.

[13] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge, LAK '13*, pages 170–179. ACM Press, 2013.

[14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM(JACM)*, 46(5):604–632, 1999.

[15] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pages 199–208. ACM Press, 2010.

[16] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, and J. X. Yu. Influence maximization over large-scale social networks: A bounded linear approach. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14*, pages 171–180. ACM Press, 2014.

[17] Q. Meng and P. J. Kennedy. Discovering influential authors in heterogeneous academic networks by a co-ranking method. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13*, pages 1029–1036. ACM Press, 2013.

[18] T. Schellens and M. Valcke. Fostering knowledge construction in university students through asynchronous discussion groups. *Computers & Education*, 46(4):349–370, 2006.

[19] A. Singh, D. P, and D. Raghu. Retrieving similar discussion forum threads: A structure based approach. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 135–144. ACM Press, 2012.

[20] G. S. Stump, J. DeBoer, J. Whittinghill, and L. Breslow. Development of a framework to classify mooc discussion forum posts: Methodology and challenges. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2013.

[21] H. Wang, C. Wang, C. Zhai, and J. Han. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11*, pages 435–444. ACM Press, 2011.

[22] M. Wen, D. Yang, and C. Rose. Sentiment analysis in mooc discussion forums: What does it tell us? In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 130–137. Chapman & Hall/CRC Press, 2014.

[23] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pages 266–275. ACM Press, 2003.

[24] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. Pagerank with priors: An influence propagation perspective. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13*, pages 2740–2746. AAAI Press, 2013.

[25] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 257–260. Chapman & Hall/CRC Press, 2014.

[26] R. Yeniterzi and J. Callan. Analyzing bias in cqa-based expert finding test sets. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14*, pages 967–970. ACM Press, 2014.

[27] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14*, pages 283–292. ACM Press, 2014.

[28] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *Proceedings of the 7th IEEE International Conference on Data Mining, ICDM '07*, pages 739–744. IEEE Press, 2007.

[29] G. Zhou, S. Lai, K. Liu, and J. Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 1662–1666. ACM Press, 2012.

[30] H. Zhu, H. Cao, H. Xiong, E. Chen, and J. Tian. Towards expert finding by leveraging relevant categories in authority ranking. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 2221–2224. ACM Press, 2011.

Modeling Learners' Social Centrality and Performance through Language and Discourse

Nia M. Dowell
Department of Psychology
Institute for Intelligent Systems
University of Memphis
365 Innovation Drive
Memphis, TN 38152
+1 901-678-5102
ndowell@memphis.edu

Arthur C. Graesser
Department of Psychology
Institute for Intelligent Systems
University of Memphis
365 Innovation Drive
Memphis, TN 38152
+1 901-678-5102
a-graesser@memphis.edu

Thieme A. Hennis
Delft Extension School
Delft University of Technology
2628 BX, Delft
+31651855220
t.a.hennis@tudelft.nl

Oleksandra Skrypnyk
School of Education
University of South Australia
Adelaide, Australia
+61 402918694
olesandra.skrypnyk@mymail.unisa.edu.au

Shane Dawson
Learning and Teaching Unit
University of South Australia
Adelaide, Australia
+61 883027850
shane.dawson@unisa.eduau

Pieter de Vries
Systems Engineering Department
Participatory Systems Design
Delft University of Technology
2628 BX Delft, Netherlands
+31651517278
pieter.devries@tudelft.nl

Srećko Joksimović
School of Interactive Arts and Technology
Simon Fraser University
Burnaby, Canada
+1604-375-2496
sjoksimo@sfu.ca

Dragan Gašević
Schools of Education and Informatics
University of Edinburgh
Edinburgh, United Kingdom
+44 131 651 6243
dragan.gasevic@ed.ac.uk

Vitomir Kovanović
Schools of Education and Informatics
University of Edinburgh
+1604-375-2496
v.kovanovic@ed.ac.uk

ABSTRACT

There is an emerging trend in higher education for the adoption of massive open online courses (MOOCs). However, despite this interest in learning at scale, there has been limited work investigating the impact MOOCs can play on student learning. In this study, we adopt a novel approach, using language and discourse as a tool to explore its association with two established measures related to learning: traditional academic performance and social centrality. We demonstrate how characteristics of language diagnostically reveal the performance and social position of learners as they interact in a MOOC. We use Coh-Metrix, a theoretically grounded, computational linguistic modeling tool, to explore students' forum postings across five potent discourse dimensions. Using a Social Network Analysis (SNA) methodology, we determine learners' social centrality. Linear mixed-effect modeling is used for all other analyses to control for individual learner and text characteristics. The results indicate that learners performed significantly better when they engaged in more expository style discourse, with surface and deep level cohesive integration, abstract language, and simple syntactic structures. However, measures of social centrality revealed a different picture. Learners garnered a more significant and central position in their social network when they engaged with more narrative style discourse with less overlap between words and ideas, simpler syntactic structures and abstract words. Implications for further research and practice are discussed regarding the misalignment between these two learning-related outcomes.

Keywords

Social Centrality, Learning, Discourse, Coh-Metrix, MOOCs

1. INTRODUCTION

Advances in educational technologies and a desire for increased access to learning, are enabling the development of pedagogical environments at scale, such as Massive Open Online Courses (MOOCs) [41]. Open online courses have the potential to advance education on a global level, by providing the masses with broader access to lifelong learning opportunities. Additionally, the insulated nature of the MOOC web-based platforms allows valuable learning dynamics to be detailed at unprecedented resolution and scale. As such, the digital traces left by learners are regarded as a gold mine that can offer powerful insights into the learning process, resulting in the advancement of educational sciences and substantially improved learning environments.

While the scale of the data has grown, making sense of data from the learning environments is not a novel effort. Prior to the arrival of MOOCs, similar endeavors were undertaken at smaller scale in the domains of computer-supported collaborative learning and intelligent tutoring systems, among others. The volume of student behavior and performance data produced in those interactions motivated the fields of educational data mining (EDM) and learning analytics (LA) [37]. Both of these research communities have leveraged this fine-grained data and aligned with educational

theory. The EDM community offer methods for exploring learners and educational settings, while LA focuses on the measurement, collection, and analyses that aim at optimizing the learning process [38]. That said, inquiring into MOOCs and other unexplored learning environments requires inputs from both communities. Direct application of methodologies, theoretical frameworks, and established analytics require deeper understanding of the relationships between parts of the whole, to enable drawing the relevant parallels with existing research.

Drawing on this, this paper adopts a novel approach, which uses language and discourse as a tool to explore its association with two established measures of learning, namely traditional academic performance and social centrality. Specifically, we are investigating the extent to which characteristics of language diagnostically reveal the performance and social position of students as they interact in a MOOC. As a methodological contribution, we adopt a theoretically grounded computational linguistics modeling approach to explore students' forum posting, within a MOOC, across five potent discourse dimensions. In line with current practice, we implement a Social Network Analysis (SNA) methodology to monitor and detect learners' social centrality. Students' performance in the course, i.e. course grade, is represented by an aggregate measure combining scores for the essays submitted during the MOOC, and a final peer-evaluated, open-ended written-assignment. Linear mixed-effects modeling approach is used for all other analyses to control for individual learner and text characteristics. This design allows us to contrast the linguistic profiles of high performing learners and centrally situated learners. Consequently, we gain insights into the qualitative differences between these two different learning-related outcomes. Finally, we explored whether the discourse features characterizing learning-related outcomes varied within different learner population, namely across all learners in the MOOC and within a subset of active learners.

The subsequent sections of the paper are organized as follows. First, we provide a brief overview of language and discourse situated within the contexts of psychological frameworks of comprehension and learning. Then, the following two sections address the traditional application of social network analysis, including theoretical foundations, as well as interpretations applied in MOOCs research. We then move into the methodological features of the current investigation, and conclude the paper with a detailed discussion of the results in the context of theory, as well as a general discussion of the theoretical, methodological, and practical implications for the EDM and LA community.

2. THEORETICAL BACKGROUND

2.1 Language and Discourse

Across academic fields, there has been a burgeoning literature demonstrating the usefulness of language and discourse in predicting a number of psychological, affective, cognitive, and social phenomena, ranging from personality to emotion to learning to successful group interactions (e.g. [6,10,26]). Within the educational contexts, there are many critical learning-related constructs that cannot be directly measured, but can be inferred from measurable signals like language and other behavioral patterns. Working with these barriers, we are continually pushing beyond the boundaries of established implementation. In that realm, it is particularly important that these endeavors be guided by established theory. A number of psychological models of discourse comprehension and learning, such as the construction-integration, constructionist, and indexical-embodiment models, lend themselves nicely to the exploration of learning related phenomena in computer-mediated educational environments. These psychological frameworks have identified the representations, structures, strategies, and processes at multiple levels of discourse [16,23,40]. Five levels have commonly been offered in these frameworks: (1) words, (2) syntax, (3) the explicit textbase, (4) the situation model (sometimes called the mental model), and (5) the discourse genre and rhetorical structure (the type of discourse and its composition). In the learning context, learners can experience communication misalignments and comprehension breakdowns at different levels. Such breakdowns and misalignments have important implications for the learning process. In this paper we adopt this multilevel approach to the analysis of language and discourse.

With regard to analytical approaches, there has been extensive knowledge gleamed from manual content analyses of learners' discourse during educational interactions, however, these methods are no longer a viable option with the increasing scale of educational data. As such, researchers have been incorporating automated linguistic analysis, including more shallow level word counts and deeper level discourse analysis approaches. Both levels of linguistic analysis are informative. Content analysis using word-counting methods allows getting a fast overview of learners' participation levels, as well as assessing specific words. For instance, a study by Wen and colleagues [43] is an example of incorporating word counts (LIWC) of theory-informed and carefully selected words with manual message coding. Their work links specific (and thus identifiable and countable) words used by the students with the degree of their engagement and commitment to remain in the course.

To extend analysis of learning-related phenomena beyond the shallow level word counts, one needs to conduct a deeper level discourse analysis employing sophisticated natural language processing techniques, e.g. syntactic parsing and cohesion computation. For example, Dowell and colleagues [11] explored the possibility of using discourse features to predict student performance during collaborative learning interactions. Their results indicated that students who engaged in deeper cohesive integration and generated more complicated syntactic structures performed significantly better. In line with this, Cade and others [3] demonstrated that cognitive linguistic cues can be used in detecting students' socio-affective attitudes towards fellow students in CMCL environments. As a whole, these studies highlight the critical and complex role of language and discourse. This is, perhaps, not surprising, since language is a primary means for expressing and communicating information in computer-mediated learning environments.

2.2 Social Network Analysis in Educational Research

Social Network Analysis (SNA) is a methodology that is increasingly being used for analyzing learning-related phenomena, especially in online settings [25]. SNA has gained popularity with researchers who view social relationships between students as an aspect influencing overall educational experience and learning outcomes (i.e. [33]). Its methodology is grounded in systematic empirical data [4:8], as well as "motivated by a relational intuition based on ties connecting social actors" (*ibid.*). Studies that employ SNA, aim at revealing the role of social relationships in learning, around such issues as *who is central in a social learning network, who is talking to whom,* and *who is participating peripherally* and *how those interaction patterns influence learning* [4,25,42]. Due to such focus, SNA provides the

theoretical and methodological tools to understand activities and social processes that students and teachers engage with. [25,31]

Traditionally, the analyses of social networks of learners have been derived from participation in discussion forums in formal online courses. The relationship between learners' position in a social network and student academic performance is well documented, in this context [5,14,33]. The general finding in this literature shows more centrally situated learners tend to get higher final grades [33]. Moreover, Russo and Koesten [34] showed that network centrality (measured as in-degree and out-degree) is a significant predictor of cognitive learning outcome. Rizzuto and others [32] found that network density significantly predicted the scores reflecting course material comprehension. Reflective of the finding from these studies a students' position in a network also influences their overall sense of community [9]. These studies suggest, in the context of formal online learning, individuals who are centrally positioned in their network perform better, and feel a stronger sense of connection than students that are more peripheral in the network structure.

In the context of MOOCs, SNA is increasingly used to explore learning-related phenomena [13]. For example, Gilliani et al. [15] applied SNA to capture broad trends in communication and the roles of individuals in facilitating discussions [15]. Another example of SNA in MOOCs is a study by Yang and colleagues [44], which suggests that learners who join forums (i.e. networks of learners) earlier are likely to persist in the course, in contrast to their counterparts who joined later and found it difficult to form social bonds. This finding is parallel to prior findings in the domain of traditional online learning revealing that learners central to the social network tend to have a higher sense of belonging to the group [8]. However, there is research that suggests the interpretation of SNA in MOOCs requires further attention. For example, the relationship between student centrality in MOOC discussion forums and their academic performance (i.e., final grade), has been shown to be context dependent [21]. Jiang and colleagues [21] demonstrated that in Algebra MOOC, betweenness and degree centrality yielded significant correlation with the final grade, while none of the metrics analyzed (i.e., closeness, degree, and betweenness centrality) was significantly correlated with the learning outcome in a Financial Planning MOOC.

Automated linguistic analysis of student interactions, within computer-mediated learning environments, can compliment SNA techniques by adding rich contextual information to the structural patterns of learner interactions. However, the combination of these two analytical methods is relatively sparse in the literature, beyond a few noteworthy exceptions [22,36]. Similar to the current work, is Joksimović and colleagues' [22] analysis of students' interaction patterns in a distributed MOOC, i.e. learner interactions take place via social media, and the course is based on connectivist pedagogy. Their findings pinpoint specific discourse features that were predictive of a learners' accumulation of social capital.

2.3 Research Questions

To summarize, SNA is a widely used tool for exploring learning processes that take place in MOOCs, largely due to its theoretical foundation and established application in formal educational contexts. However, given the open nature of scaled online courses, the interpretation of SNA in MOOCs requires further attention. This study approaches language as the primary means for communication and a window into inferring learning-related phenomena. We apply discourse analysis as a proxy for providing qualitative information about the position of learners in the network and their performance. The analysis focuses around the following research questions: Which characteristics of language diagnostically reveal the performance and social position of students as they interact in a MOOC? And do these features operate similarly with different learner populations, namely across all learners in a MOOC and within a subset of active learners?

3. METHODS

3.1 Participants

The study analyzed forum discussion posted on the edX platform, within the course NGI101x Next Generation Infrastructures (NGIx). It ran for 8 weeks in the period of April 22 – July 8, 2014. The subject area of the analyzed MOOC fell under the domain of applied non-life soft sciences [2]; the course objective was to introduce the complexity of infrastructure systems, familiarize students with the main concepts within the area, as well as with the practical approaches to the infra-systems analysis. In total 16,091 participants enrolled and 517 received certificate of completion (passed). To pass the course the students needed to receive a score of 0.7 (out of 1) or higher. The grade was derived from the submission of 3-6 open-ended papers (60% of the grade) and a final issue paper (40% of the grade) that was peer assessed by several co-learners. The dataset for the analysis in this study included 1,754 participants (N_{post}=7,244, M=4.13, SD=9.85, $Q1$=1.0, $Q3$=4.0, Min=1.0, Max=180), i.e. all those who used the course forum. Forum data was collected from the edX platform in the JSON format, and included all the information specified within the edX discussion forums data documentation[1].

3.2 Analyses

3.2.1 Social Network Analysis

Although other approaches have been proposed, the most common approach for extracting social networks from online discussions is to consider each message as directed to the previous one in the thread [25,31]. In the current study, we followed the approach suggested in [24,25,31], among others. Specifically, social graph representing interaction within the discussion forum included all the students who posted a message(s). For example, author A1 initiated the discussion, and author A2 posted a message directly into the thread, in reply to A1's initial thread message, we would add directed edge A2->A1. Then, if author A3 replied to the message posted by author A2, we would include a direct edge A3->A2 to the graph. If author A4 started a nested discussion as a reply to A1's initial post, then A4 would have a direct edge to A1. The concept of centrality has been commonly used to assess the importance of an individual node within a social network [12,42]. The following well-established SNA measures [42], that capture various notions of a graph structural centrality, were calculated for each learner in the social network extracted:

- **Degree Centrality** – the number of edges a node has in a network;
- **Closeness Centrality** – the distance of an individual node in the network from all the other nodes;
- **Betweenness Centrality** – the number of shortest paths between any two nodes that pass via a given node.

Degree centrality is generally used to capture the "potential for activity in communication" [12:219] or the *popularity* [31] of a node in a social network. Betweenness centrality, on the other hand, represents a *potential for influence* over the information

[1] http://devdata.readthedocs.org/en/latest/internal_data_formats/discussion_data.html

flow, as it *bridges* the parts of the network that were disconnected otherwise [12,31,42]. Finally, the concept of closeness centrality refers to the distance between a learner and the other participants of the network. In a MOOC, closeness centrality can be interpreted as the extent to which a learner is in the middle of what is happening on the forum. The relationship between students' linguistic properties and their position in the social network, measured through the three properties described above, has been investigated in this study. The social network variables were analyzed using *igraph 0.7.1* [7], a comprehensive R software package for complex social network analysis research.

3.2.2 Coh-Metrix Analyses

Prior to Coh-Metrix analyses, the logs were cleaned and parsed to facilitate a student level evaluation. Thus, text files were created that included all contributions from a single learner, yielding a total of 1,754 text files, one for each student. All files were then analyzed using Coh-Metrix. Coh-Metrix (www.cohmetrix.com) is a computational linguistics facility that provides measures of over 100 measures of various types of cohesion, including co-reference, referential, causal, spatial, temporal, and structural cohesion [18,26]. Coh-Metrix also has measures of linguistic complexity, characteristics of words, and readability scores. Currently, Coh-Metrix is being used to analyze texts in K-12 for the Common Core standards and states throughout the U.S. More than 50 published studies have demonstrated that Coh-Metrix indices can be used to detect subtle differences in text and discourse [26].

There is a need to reduce the large number of measures provided by Coh-Metrix into a more manageable number of measures. This was achieved in a study that examined 53 Coh-Metrix measures for 37,520 texts in the TASA (Touchstone Applied Science Association) corpus, which represents what typical high school students have read throughout their lifetime [17]. A principal components analysis was conducted on the corpus, yielding eight components that explained an impressive 67.3% of the variability among texts; the top five components explained over 50% of the variance. Importantly, the components aligned with the language-discourse levels previously proposed in multilevel theoretical frameworks of cognition and comprehension [16,23,40]. These theoretical frameworks identify the representations, structures, strategies, and processes at different levels of language and discourse, and thus are ideal for investigating trends in learning-oriented conversations. Below are the five major dimensions, or latent components, that may be useful for understanding trends in learning-oriented, but inherently social, conversations:

- **Narrativity.** The extent to which the text is in the narrative genre, which conveys a story, a procedure, or a sequence of episodes of actions and events with animate beings. Informational texts on unfamiliar topics are at the opposite end of the continuum.
- **Deep Cohesion.** The extent to which the ideas in the text are cohesively connected at a deeper conceptual level that signifies causality or intentionality.
- **Referential Cohesion.** The extent to which explicit words and ideas in the text are connected with each other as the text unfolds.
- **Syntactic Simplicity.** Sentences with few words and simple, familiar syntactic structures. At the opposite pole are structurally embedded sentences that require the reader to hold many words and ideas in working memory.
- **Word Concreteness.** The extent to which content words that are concrete, meaningful, and evoke mental images as opposed to abstract words.

3.2.3 Data Preparation

The students' performance, linguistic and network data were merged to facilitate subsequent statistical analyses. Following this, the scores were centered and normalized by removing any outliers. Specifically, the normalization procedure involved Winsorising the data based on each variable's upper and lower percentile. Finally, we were interested in exploring whether the discourse features characterizing learning-related outcomes varied within different learner population, namely across all learners in the MOOC and within a subset of active learners. To enable this analysis, we created two datasets. The *All Learner* dataset contained data for the full 1,754 students that participated in the MOOC. We operationalized active students as those learners who made 4 or more posts in the MOOC. The cut-off point was chosen because the top 25% of learners made 4 or more posts. The resulting *Active Learner* dataset contained the data for those top 471 learners.

3.2.4 Statistical analyses

A mixed-effects modeling approach was adopted for all analyses due to the structure of the data (e.g., inter-individual and word count variability) [30]. Mixed-effects models include a combination of fixed and random effects and can be used to assess the influence of the fixed effects on dependent variables after accounting for any extraneous random effects. The primary analyses focused on identifying the association between the discourse features, namely, Narrativity, Deep Cohesion, Referential Cohesion, Syntax Simplicity, and Word Concreteness and the learning outcomes, measured through learners' social centrality and grades. Therefore, we identified two sets of dependent measures in the present analyses: (1) learners' social centrality (Closeness, Degree, and Betweenness) and (2) learners' performance in the course (the final grade). The independent variables in all models were the five discourse features of interest.

Additionally, the influence of language on learning and social capital might vary depending on relevant learner characteristics. For instance, discourse may play a more meaningful role, for student performance and social position in a network, for more active learners than less active learners [25]. This would be in line with Gillani and others [15] conclusion that suggests the social network extracted from the learner interactions "was a noise-corrupted version of the "true" network" (p.2). Thus, we decided to further refine our analysis and create social graph only for those learners who actively participated in discussions (for the cut-off point see Section 4.2). This resulted in an additional four models, labeled as *Active Learners*, exploring the influence of language on learners' social centrality (three models) and performance (one model) for the most active participants in the course.

It is important to note that in addition to constructing the models with the five discourse features as fixed effects, *null models* with the random effects (*learner* and *word count*) but no fixed effects were also constructed. A comparison of the null random effects only model with the fixed-effect models allows us to determine whether discourse predicts social centrality and performance above and beyond the random effects. Akaike Information Criterion (AIC), Log Likelihood (LL) and a likelihood ratio test were used to determine the best fitting and most parsimonious model. In addition, we also estimate effect sizes for each model, using a pseudo R^2 method, as suggested by Nakagawa and Schielzeth [28]. For mixed-effects models, R^2 can be characterized into two varieties: marginal R^2 and conditional R^2. Marginal R^2 is associated with variance explained by fixed factors, and conditional R^2 is can be interpreted as the variance explained

by the entire model, namely random and fixed factors. Both marginal (R^2_m) and conditional (R^2_c) R^2 convey unique and relevant information regarding the model fit and variance explained, and so we report both here. The lme4 package in R [1] was used to perform all the required computation.

4. RESULTS AND DISCUSSION

4.1 Discourse and Learning

First, we assessed the relationship between learners discourse patterns and performance in the MOOC. The likelihood ratio tests indicated that both the *All Learner* and *Active Learner* models yielded a significantly better fit than the null model with $\chi^2(5) = 82.57$, $p = .001$, $R^2_m = .05$, $R^2_c = .93$, and $\chi^2(5) = 85.44$, $p = .001$, $R^2_m = .21$, $R^2_c = .95$, respectively. A number of conclusions can be drawn from this initial model fit evaluation and inspection of R^2 variance. First, the model comparisons imply that the discourse features were able to add a significant improvement in predicting the learners' performance above and beyond individual participant characteristics. Second, for the *All Learner* model, discourse and individual participant features explained about 93% of the predictable variance, with 5% of the variance being accounted for by the discourse features. However, the discourse features alone were able to explain a total of 21% of predictable variance in active learners' performance. The observed difference in variance suggests discourse features are more accurate at predicting active learners' performance than that of learners who are less active in the course. It is important to note that the difference in the explained variance for the *All Learner* and *Active Learner* models is not a result of the students simply being more prolific, because we controlled for number of words. Instead the findings might be reflecting a more substantive difference for the active students' potency of thought integration, complexity and communication style, beyond the observation that they are communicating more, compared to the overall learner population. Table 1 shows the discourse features that were predictive of learning performance for both the *All Learner* and *Active Learner* models. As can be seen from Table 1, all five levels of discourse were predictive of learning performance for the *All Learner* models, and four of the five levels were predictive of learning in the *Active Learner* models. Specifically, learners who engaged in more expository style discourse with referential and deep level cohesive integration, abstract language, and simple syntactic structures performed significantly better in the course.

Narrative discourse expresses events and actions performed by characters that unfold over time, as is typical in everyday oral communication, folktales, drama, and short stories [35]. In contrast to narrative, expository language is decontextualized and generally informs the audience about new concepts, broad truths, and technical material as in the case of academic articles and college textbooks. The genre of a text can be particularly revealing with regard to its difficulty. For example, narrative text is substantially easier to read, comprehend, and recall than informational or expository text [16]. From a constructionist theory [19,20] view, this is because expository discourse frequently presents abstract categories and less familiar information that require learners to have extensive background knowledge about the topics in order to generate the inferences necessary for comprehension [39]. As a reminder, our measure of narrativity/expository is a single continuum, wherein higher numbers indicate narrative style discourse and lower numbers indicate expository style discourse. Thus, the negative findings for Narrativity (Table 1) can be extrapolated to conclude that learners who articulated their responses in a more expository style, mirroring the informational nature of their class material, extracted enough information about the subject to generate inferential processing. Such interpretation is in line with other research showing knowledgeable students develop more comprehensive representations from material than less knowledgeable students [27], and can inferentially relate the information they derive from text better than readers with less background knowledge.

In line with Kintsch's [23] construction-integration theory, Coh-Metrix distinguishes between multiple types of cohesion which fall under two main forms, namely textbase (i.e. referential cohesion) and situation model cohesion (i.e. deep cohesion). Referential or textbase cohesion is primarily maintained through the bridging devices, i.e. the overlap in words, or semantic references. In this context, the findings for referential cohesion suggest that learners who perform better, construct their messages using more bridging devices

A theory of situation model cohesion has been described by [45] that characterizes it as knowledge elaborations that are product of incorporating information derived from the explicit texts with background world knowledge. Coh-Metrix analyzes the situation model dimension on causation, intentionality, space, and time [26]. With regard to the findings for deep cohesion, this suggests that students who are learning are engaging in deeper integration of topics with their background knowledge, generating more inferences to address any conceptual and structural gaps, and consequentially increasing the probability of comprehension. The results for syntax show that simple syntactic structures were associated with better performance. However, this finding was not significant in the *Active Learner* model.

Table 1. Descriptive Statistics and Mixed-Effects Model Coefficients for Predicting Performance with Language

Measure	All Learner Model				Active Learner Model			
	M	***SD***	**β**	***SE***	***M***	***SD***	**β**	***SE***
Narrativity	0.00	1.00	-.20**	.02	-0.23	0.69	-.60**	.07
Deep Cohesion	0.00	1.00	.08**	. 02	0.27	0.55	.19*	.08
Referential Cohesion	0.00	1.00	.08**	. 02	-0.26	0.64	.35**	.07
Syntax Simplicity	0.00	1.00	.07**	. 02	0.36	0.67	.08	.07
Word Concreteness	0.00	1.00	-.13**	. 02	-0.25	0.51	-.35**	.09

Note: * $p < .05$; ** $p < .001$. Mean (***M***). Standard deviation (***SD***). Fixed effect coefficient (**β**). Standard error (*SE*). All Learner Model *N*= 1754, Active Learner Model *N*= 471.

Coh-Metrix measures psychological dimensions of words that influence language complexity. As a reminder, our measure of word concreteness is a single continuum, wherein scores are higher when a higher percentage of the content words are concrete, are meaningful, and evoked mental images – as opposed to being abstract. Thus, the negative findings for word concreteness show learners who engaged using more abstract language performed significantly better in the course. There are interesting interpretations from the view of Petty and Cacioppo's Elaboration Likelihood Model (ELM) [29]. The ELM outlines several factors that affect both the ability and motivation to elaborate on arguments contained in messages. If ability to process is impaired, or motivation to process is low, the elaboration and thought density of the learners' communication

would likely suffer. With the exception of syntax ease, the findings suggest students who adopt central route linguistic characteristics perform significantly better than those who use peripheral linguistic features.

4.2 Discourse and Social Centrality

Next, we investigated the relationship between learners' discourse patterns and their position in the social network. The likelihood ratio tests indicated that the *All Learner* models for Closeness, Betweenness and Degree yielded a significantly better fit than the null random effects only models with $\chi^2(5) = 135.74$, $p = .001$, $R^2_m = .07$, $R^2_c = .93$, $\chi^2(5) = 25.63$, $p = .0001$, $R^2_m = .01$, $R^2_c = .91$, and $\chi^2(5) = 62.19$, $p = .0001$, $R^2_m = .02$, $R^2_c = .94$, respectively. Similarly, for the *Active Learner* models, the likelihood ratio tests indicated that Closeness, Betweenness and Degree yielded a significantly better fit than the null models with $\chi^2(5) = 38.39$, $p = .0001$, $R^2_m = .08$, $R^2_c = .94$, $\chi^2(5) = 45.92$, $p = .0001$, $R^2_m = .09$, $R^2_c = .94$, and $\chi^2(5) = 63.78$, $p = .0001$, $R^2_m = .12$ and $R^2_c = .96$, respectively. Similar to the results for performance, the model comparisons imply that the discourse features were able to add a significant improvement in predicting the learners' social centrality above and beyond participant characteristics. In line with this, across the three *All Learner* models, our features explained about 92% of the predictable variance, with 10% of the variance being accounted for by the linguistic features. However, the discourse features were able to explain a total of 29% of predictable variance in active learners' social centrality. Again, this suggests discourse more accurately predicts active learners' position than less active learners. The details of the *All Learner* and *Active Learner* models are reported in Table 2 and Table 3. Interestingly, the pattern of discourse features associated with learners' social centrality differed from the one observed for students' performance in the MOOC. Instead, learners who garnered central positions in the network engaged in narrative discourse with lower referential cohesion, abstract words and simple syntactic structures. With the exception of word abstractness, this pattern is indicative of informal communication.

Across all learners, higher closeness centrality is characterized by more narrative style discourse with less overlap between words and ideas (i.e. low referential cohesion), simple syntactic structures and abstract words. For active learners, the pattern is similar, with only narrativity and referential cohesion being significant. The conventional interpretation of closeness centrality indicates the efficiency of an individual in passing the information directly onto all other individuals in the social network [12]. Due to the nature of MOOC centralized forums, it can be inferred that shorter distance to all the learners can be obtained, if the individual participates in many various discussion threads. Therefore, individuals who are more active and initiate more topical messages yielding replies from many other learners, or reply to many other discussions, would use language characterized by simpler structures, narrative style, and lower referential cohesion. Similar pattern for higher narrativity and lower referential cohesion has been observed in the discourse of learners with high degree and betweenness centrality in a distributed MOOC – a course where learner interactions take place on social media, rather than on the course platform [22]. Although conventionally betweenness centrality is associated with the brokering of information between sub-groups, this is questionable in the context of an online open centralized discussion forum.

These results suggest that learners who attained a more prominent social centrality position used more conversational style discourse. Most noteworthy is that these results do not mirror the pattern observed for high performing learners. On the contrary, linguistic profiles of high performing learners are characterized by formal discourse that uses expository style language (i.e. negative relationship with narrativity), and more surface and deep level cohesive integration (i.e. positive relationship with referential and deep cohesion) (Table 1).

Table 2. All Learner Mixed-Effects Model Coefficients for Predicting Social Network Centrality with Language

Measure	Closeness		Betweenness		Degree	
	β	***SE***	**β**	***SE***	**β**	***SE***
Narrativity	.070*	.03	.03	.03	.07**	.02
Deep Cohesion	.008	.02	.01	.02	-.02	.02
Referential Cohesion	-.15**	.03	-.02	.03	-.06**	.02
Syntax Simplicity	.13**	.03	.09*	. 03	.06*	.02
Word Concreteness	-.09**	.03	-.03	. 02	-.05*	.02

Note: * $p < .05$; ** $p < .001$. Mean (***M***). Standard deviation (***SD***). Fixed effect coefficient (**β**). Standard error (*SE*). *N*= 1754.

Table 3. Active Learner Mixed-Effects Model Coefficients for Predicting Social Network Centrality with Language

Measure	Closeness		Betweenness		Degree	
	β	***SE***	**β**	***SE***	**β**	***SE***
Narrativity	.32**	.07	.17*	.07	.21**	.06
Deep Cohesion	-.06	.08	.02	.08	.05	.08
Referential Cohesion	-.33**	.07	.11	.07	.09	.07
Syntax Simplicity	.07	.07	.42**	.07	.47**	.07
Word Concreteness	.14	.09	-.07	.09	-.06	.09

Note: * $p < .05$; ** $p < .001$. Mean (***M***). Standard deviation (***SD***). Fixed effect coefficient (**β**). Standard error (*SE*). *N*= 471.

5. GENERAL DISCUSSION

This paper adopted a novel approach, which uses language and discourse as a tool to explore its association with two established measures of learning, namely traditional academic performance and social centrality. Specifically, we explored the extent to which characteristics of discourse diagnostically reveal the performance and social position of learners as they interact in a MOOC. The findings present some methodological, theoretical, and practical implications for the educational data mining and learning analytics communities. First, as a methodological contribution, we have highlighted the rich contextual information that can be gleaned from combing deeper level linguistic analysis and SNA. Particularly, discourse features add a significant improvement in predicting both the performance and social network positioning in MOOC forums.

Secondly, the results pose some important theoretical and practical implications for transferring analytic approaches to scaled environments without careful consideration. The results indicate that learners who performed significantly better engaged in more expository style discourse, with surface and deep level cohesive integration, abstract language, and simple syntactic structures. However, linguistic profiles of the centrally positioned learners differed from the high performers. Learners with a more significant and central position in their communication network engaged using a more narrative style discourse with less overlap between words and ideas, simpler syntactic structures and abstract words. In other words, high performers and those with central

positions in the network are not necessarily the same individuals. The misalignment between the linguistic features associated with improved performance and more centrally located network positions is captured by the discrepant pattern for narrative, referential and deep cohesion. These three discourse features are inversely related with high performance and centrality in networks. This difference has important implications because these linguistic dimensions are strongly associated with comprehension according to construction-integration and constructivist theories.

The study also suggests that in open online environments two established measures of learning: traditional academic performance and social centrality reflect different learning outcomes. Academic performance represents a snapshot of students' mastery of the subject, and is one way of accessing the state of subject comprehension. Positioning in social network represents a snapshot of the participation processes and social learning activities. In this study, we demonstrate that the skills associated with these two learning-related outcomes differ.

It could be speculated that the observed misalignment between linguistic performance and social network position in the analyzed open online course, shows the difference in communication patterns of formal and informal learning environments. Formal learning environments have a clearer start and end, and often require participation related to the subject matter, as embedded in tasks, or course design. In open learning environments, adult learners can opt in and opt out of the learning situations. The issue is further complicated by the discussions being held by the learners on MOOC forums on various topics: from subject matter, to technical troubleshooting, or clarification of administrative issues. Centralized forums of MOOCs are more than a social learning space; they are also a communication space. As a result, learners' high activity on a number of issues during one or two weeks of the course may result in a more central position in the network of learners, but may not necessarily indicate that the learners engaged with the content, or demonstrated the required understanding of the subject at the end of the course.

It is unclear from this study what relationship should be deduced between learning and social centrality measures within in the open online environments. At the minimum, the findings suggest that the social positioning in a network of learners in a MOOC may not be equivalent with measured academic performance. Further research is needed to understanding what analytical approaches, such as SNA, are reflecting in emerging educational environments.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (REC 0106965, ITR 0325428, HCC 0834847) and (DRK-12-0918409), the Institute of Education Sciences (R305G020018, R305A080589), The Gates Foundation, U.S. Department of Homeland Security (Z934002/UTAA08-063), Natural Sciences and Engineering Research Council of Canada (356029), Social Sciences and Humanities Research Council of Canada (435-2013-1708), and Canada Research Chairs Program. Any opinions, findings, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

7. REFERENCES

[1] Bates, D., Maechler, M., Bolker, B., et al. *lme4: Linear mixed-effects models using Eigen and S4.* 2014.

[2] Biglan, A. The characteristics of subject matter in different academic areas. *Journal of Applied Psychology 57*, (1973), 195–203.

[3] Cade, W.L., Dowell, N.M., Graesser, A.C., Tausczik, Y.R., and Pennebaker, J.W. Modeling student socioaffective responses to group interactions in a collaborative online chat environment. In J. Stamper, Z. Pardos, M. Mavrikis and B.M. McLaren, eds., *Proceedings of the 7th International Conference on Educational Data Mining.* Springer, Berlin, 2014, 399–400.

[4] Carolan, B.V. *Social Network Analysis Education: Theory, Methods & Applications.* SAGE Publications, Inc. SAGE Publications, Inc., 2014.

[5] Cho, H., Gay, G., Davidson, B., and Ingraffea, A. Social networks, communication styles, and learning performance in a CSCL community. *Computers & Education 49*, 2 (2007), 309–329.

[6] Chung, C.K. and Pennebaker, J.W. Using Computerized Text Analysis to Track Social Processes. In T. Holtgraves, ed., *Oxford Handbooks Online*. Oxford, 2014.

[7] Csardi, G. and Nepusz, T. The igraph Software Package for Complex Network Research. *InterJournal Complex Systems*, (2006), 1695.

[8] Dawson, S. Online forum discussion interactions as an indicator of student community. *Australasian Journal of Educational Technology 22*, 4 (2006), 495–510.

[9] Dawson, S. A study of the relationship between student social networks and sense of community. *Educational Technology & Society 11*, 3 (2008), 224–238.

[10] D'Mello, S. and Graesser, A.C. Language and Discourse Are Powerful Signals of Student Emotions during Tutoring. *IEEE Transactions on Learning Technologies 5*, 4 (2012), 304–317.

[11] Dowell, N.M., Cade, W.L., Tausczik, Y.R., Pennebaker, J.W., and Graesser, A.C. What works: Creating adaptive and intelligent systems for collaborative learning support. In S. Trausan-Matu, K.E. Boyer, M. Crosby and K. Panourgia, eds., *Twelfth International Conference on Intelligent Tutoring Systems.* Springer, Berlin, 2014, 124–133.

[12] Freeman, L.C. Centrality in social networks conceptual clarification. *Social networks 1*, 3 (1979), 215–239.

[13] Gasevic, D., Kovanovic, V., Joksimovic, S., and Siemens, G. Where is research on massive open online courses headed? A data analysis of the MOOC Research Initiative. *The International Review of Research in Open and Distributed Learning 15*, 5 (2014).

[14] Gašević, D., Zouaq, A., and Janzen, R. "Choose Your Classmates, Your GPA Is at Stake!": The Association of Cross-Class Social Ties and Academic Performance. *American Behavioral Scientist*, (2013).

[15] Gillani, N., Yasseri, T., Eynon, R., and Hjorth, I. Structural limitations of learning in a crowd: communication vulnerability and information diffusion in MOOCs. *Scientific reports 4*, (2014).

[16] Graesser, A.C. and McNamara, D.S. Computational Analyses of Multilevel Discourse Comprehension. *Topics in Cognitive Science 3*, 2 (2011), 371–398.

[17] Graesser, A.C., McNamara, D.S., and Kulikowich, J.M. Coh-Metrix: Providing Multilevel Analyses of Text Characteristics. *Educational Researcher 40*, 5 (2011), 223–234.

[18] Graesser, A.C., McNamara, D.S., Louwerse, M.M., and Cai, Z. Coh-metrix: analysis of text on cohesion and language. *Behavior research methods, instruments, & computers: a journal of the Psychonomic Society, Inc 36*, 2 (2004), 193–202.

[19] Graesser, A.C., Singer, M., and Trabasso, T. Constructing Inferences during Narrative Text Comprehension. *Psychological Review 101*, 3 (1994), 371–95.

[20] Graesser, A.C. and Wiemer-Hastings, K. Situation models and concepts in story comprehension. In S.R. Goldman, A.C. Graesser and P. van den Broek, eds., *Narrative comprehension, causality, and coherence*. Mahwah, NJ, 1999, 77–92.

[21] Jiang, S., Fitzhugh, S.M., and Warschauer, M. Social Positioning and Performance in MOOCs. Proceedings of the Workshops held at Educational Data Mining 2014, co-located with 7th International Conference on Educational Data Mining (EDM 2014), (2014), 14.

[22] Joksimović, S., Dowell, N.M., Skrypnyk, O., et al. How do you connect? Analysis of Social Capital Accumulation in connectivist MOOCs. In *Proceedings from the 5th International Learning Analytics and Knowledge (LAK) Conference*. Poughkeepsie, New York, 2015.

[23] Kintsch, W. *Comprehension: A Paradigm for Cognition.* Cambridge University Press, Cambridge, U.K., 1998.

[24] Kovanović, V., Joksimović, S., Gašević, D., and Hatala, M. What is the Source of Social Capital? The Association between Social Network Position and Social Presence in Communities of Inquiry. *Proceedings of the Workshops held at Educational Data Mining 2014, (EDM 2014)*, (2014), 1–8.

[25] De Laat, M., Lally, V., Lipponen, L., and Simons, R.-J. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for Social Network Analysis. *International Journal of Computer-Supported Collaborative Learning 2*, 1 (2007), 87–103.

[26] McNamara, D.S., Graesser, A.C., McCarthy, P.M., and Cai, Z. *Automated evaluation of text and discourse with Coh-Metrix.* Cambridge University Press., Cambridge, M.A., 2014.

[27] McNamara, D.S., Kintsch, E., Songer, N.B., and Kintsch, W. Are Good Texts Always Better? Interactions of Text Coherence, Background Knowledge, and Levels of Understanding in Learning from Text. *Cognition and Instruction 14*, 1 (1996), 1–43.

[28] Nakagawa, S. and Schielzeth, H. A general and simple method for obtaining R2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution 4*, 2 (2013), 133–142.

[29] Petty, R.E., Cacioppo, J.T., Strathman, A.J., and Priester, J.R. To Think or Not to Think: Exploring Two Routes to Persuasion. In T.C. Brock and M.C. Green, eds., *Persuasion: Psychological insights and perspectives, 2nd ed*. Sage Publications, Inc, Thousand Oaks, CA, US, 2005, 81–116.

[30] Pinheiro, J.C. and Bates, D.M. *Mixed-effects models in S and S-Plus.* Springer, 2000.

[31] Rabbany k., R., Takaffoli, M., and Zaïane, O.R. Social Network Analysis and Mining to Support the Assessment of On-line Student Participation. *SIGKDD Explor. Newsl. 13*, 2 (2012), 20–29.

[32] Rizzuto, T., LeDoux, J., and Hatala, J. It's not just what you know, it's who you know: Testing a model of the relative importance of social networks to academic performance. *Social Psychology of Education 12*, 2 (2009), 175–189.

[33] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education 68*, (2013), 458–472.

[34] Russo, T.C. and Koesten, J. Prestige, centrality, and learning: A social network analysis of an online class. *Communication Education 54*, 3 (2005), 254–261.

[35] Sanford, A.J. and Emmott, C. *Mind, Brain and Narrative.* Cambridge University Press, Cambridge, 2012.

[36] Scholand, A.J., Tausczik, Y.R., and Pennebaker, J.W. Assessing Group Interaction with Social Language Network Analysis. In S.-K. Chai, J.J. Salerno and P.L. Mabry, eds., *Advances in Social Computing*. Springer Berlin Heidelberg, 2010, 248–255.

[37] Siemens, G. and Baker, R.S. Learning analytics and educational data mining: towards communication and collaboration. *Proceedings of the 2nd international conference on learning analytics and knowledge*, ACM (2012), 252–254.

[38] Siemens, G. and Gašević, D. Special Issue on Learning and Knowledge Analytics. *Educ Technol Soc 15*, 3, 1–2.

[39] Singer, M. and O'Connell, G. Robust inference processes in expository text comprehension. *European Journal of Cognitive Psychology 15*, 4 (2003), 607–631.

[40] Snow, C.E. Reading for Understanding: Toward a Research and Development Program in Reading Comprehension. Rand Corporation, Santa Monica, CA, 2002.

[41] Walsh, T. and Bowen, W.G. Unlocking the Gates: How and Why Leading Universities Are Opening Up Access to Their Courses. Princeton University Press, Princeton, 2011.

[42] Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, Cambridge ; New York, 1994.

[43] Wen, M., Yang, D., and Rose, C. Linguistic Reflections of Student Engagement in Massive Open Online Courses. In *Proceedings 14th International Conference on Web and Social Media*. AAAI, Ann Arbor, MI, 2014, 525–534.

[44] Yang, D., Wen, M., Kumar, A., Xing, E., and Rose, C. Towards an Integration of Text and Graph Clustering Methods as a Lens for Studying Social Interaction in MOOCs. *The International Review of Research in Open and Distributed Learning 15*, 5 (2014).

[45] Zwaan, R.A. and Radvansky, G.A. Situation models in language comprehension and memory. *Psychological Bulletin 123*, 2 (1998), 162–185.

You are Your Words: Modeling Students' Vocabulary Knowledge with Natural Language Processing Tools

Laura K. Allen
Tempe, AZ, USA
Arizona State University
LauraKAllen@asu.edu

Danielle S. McNamara
Tempe, AZ, USA
Arizona State University
Danielle.McNamara@asu.edu

ABSTRACT
The current study investigates the degree to which the lexical properties of students' essays can inform stealth assessments of their vocabulary knowledge. In particular, we used indices calculated with the natural language processing tool, TAALES, to predict students' performance on a measure of vocabulary knowledge. To this end, two corpora were collected which contained essays from early college and high school students, respectively. The lexical properties of these essays were then calculated using TAALES. The results of this study indicated that two of the linguistic indices were able to account for 44% of the variance in the college students' vocabulary knowledge scores. Additionally, the significant indices from this first corpus analysis were able to account for a significant portion of the variance in the high school students' vocabulary scores. Overall, these results suggest that natural language processing techniques can inform stealth assessments and help to improve student models within computer-based learning environments.

Keywords
Intelligent Tutoring Systems, writing, Natural Language Processing, feedback

1. INTRODUCTION
Writing is a complex cognitive and social process that is important for both academic and professional success [1]. As contemporary societies grow increasingly reliant on text sources to communicate ideas (e.g., emails, text messages, online reports, blogs), the importance of developing proficiency in this area is more important than ever. Unfortunately, acquiring writing skills is no simple task – as evidenced by the many students who underachieve each year on national and international assessments of writing proficiency [1, 2, 3, 4]. Indeed, this text production process is complex and relies on the development of both lower and higher-level knowledge and skills, ranging from a strong knowledge of vocabulary to the strategies necessary for tying their ideas together [5, 6, 7].

To develop the skills that are required to produce high-quality texts, students need to be provided with comprehensive instruction that targets their individual strengths and weaknesses. In particular, this instruction should explicitly describe and demonstrate the skills and strategies that will be necessary during each of the phases of the writing process. Additionally, it should offer students opportunities to receive summative and formative feedback on their work, while engaging in deliberate practice. This form of *deliberate* practice is an important factor in students' development of strong writing skills [8, 9], because it can promote self-regulation of the planning, generation, and reviewing processes [9]. Unfortunately, however, deliberate practice inherently relies on individualized writing feedback. This is often difficult for teachers to provide, as they are faced with large class sizes and do not have the time to provide thorough comments on every essay that a student writes.

As a result of these classroom needs, researchers have developed computer-based writing systems that can provide students with feedback on their writing [10]. These systems have been used for both classroom assignments and high-stakes writing assessments to ease the burden of individualized essay scoring [11]. Specifically, *automated essay scoring* (AES) systems evaluate the linguistic properties of students' essays to assign them holistic scores [12, 13]. These systems use a multitude of natural language processing (NLP) and machine learning methodologies to provide these essay scores, and previous research suggests that they are often comparable to human raters [11, 13, 14, 15].

To provide students with greater context for the scores on their essays, AES systems are commonly incorporated into educational learning environments, such as *automated writing evaluation* (AWE) systems [16, 17] and *intelligent tutoring systems* (ITSs) [18]. These systems not only provide students with summative feedback on their essays (i.e., holistic scores), they also provide formative feedback and writing instruction. In order to be successful, these systems must contain algorithms that can provide individualized feedback that is relevant to students' individual skills.

Importantly, these computer-based writing environments rely on linguistic features to assess the *quality* of the individual essays submitted to the systems. Although the scores are generally valid and reliable, the systems rarely consider student-level information (e.g., their knowledge, skills, or affect) when providing feedback based on these scores. This can pose critical problems when developing adaptive components for the systems. As an example, consider two students, Mary and John, who both write essays that receive holistic scores of "3" from an AWE system. While Mary is able to clearly argue her point in the thesis and topic sentences,

her essay is weakened by simplistic language and sentence constructions. John, on the other hand, employs sophisticated vocabulary and eloquent sentences throughout his essay; however, he does a poor job of explaining his position on the argument. In this example, both students received the same score from the system; however, their essays were affected by different student-level strengths and weaknesses. Mary may have suffered from lower vocabulary knowledge and general language skills, whereas John may not have developed adequate planning and organization strategies.

One way to accommodate these individual differences is to develop user models based on students' characteristics, beyond simply their scores on essays. These models can provide more specific instruction and feedback that are tailored to students' strengths and weaknesses. One individual difference that may be particularly important to consider in these student models is *vocabulary knowledge.* Previous studies have shown that vocabulary knowledge plays a major role in the writing process, as it is strongly correlated with the scores assigned to students' essays [5, 19]. In the current paper, we examine the efficacy of NLP techniques to inform stealth assessments of this knowledge. In particular, we examine whether the lexical properties of students' essays can accurately model their scores on a standardized measure of vocabulary knowledge. Ultimately, our aim is to use these measures to provide more individualized tutoring to student users.

1.1 Stealth Assessments

In order to provide a more personalized learning experience (e.g., individualized instruction and feedback), computer-based learning environments must rely on repeated assessments of performance as students interact with the system. These measures can provide important information about students' knowledge states and learning trajectories, which can help to increase the adaptivity of these systems. Despite the importance of these assessments, however, they are not particularly conducive to robust student learning. In particular, constantly exposing students to questionnaires and tests can disrupt their learning flow [20] and subsequently harm their performance on later tasks.

As a response to this assessment problem, researchers have placed an emphasis on the development of methods that can accumulate information about student users without persistently disrupting the learning task [20, 21]. In particular, researchers have proposed the development of *stealth assessments*. These assessments are intended to measure students' performance and knowledge without requiring any explicit testing. Typically, these stealth assessments are embedded within the learning task itself and, as a result, are not able to be detected by students [22].

Within the context of computer-based learning environments, these stealth assessments can be informed by a wealth of information that can be easily logged in the system. These data can range from the speed at which someone is typing to the trajectories of their mouse movements. Snow and colleagues (2014), for example, developed stealth assessments of agency within a reading comprehension tutoring system [23]. They found that students who exhibited more systematic patterns of behavior in the system produced higher quality self-explanations compared to students who were more disordered in their choice patterns. They stated that this measure of behavior patterns could serve as a stealth assessment of agency in adaptive learning environments. Overall, stealth assessments can serve as a viable solution to the assessment problem, as they can be informed by a wide variety of data types to model the characteristics of student users (e.g., their skills, attitudes, etc.) [23, 24].

Importantly, after they have been developed, these stealth assessments can be used to enhance student models. Models of students' performance and attitudes are typically embedded in ITSs as a means to provide more individualized instruction and feedback [25]. In these systems, student users are represented by continuously updating models that are representative of their own knowledge and performance in the system. Thus, once the system has the ability to reliably assess students' particular skill sets, it can adapt in precise ways that can enhance the overall efficacy of the instruction [26].

1.2 Natural Language Processing

Natural language processing (NLP) tools provide a means through which researchers can develop stealth assessments of student characteristics [24]. In addition, these tools can help researchers to investigate the relationships between individual differences and the learning process at a more fine-grained size. By calculating indices related to multiple levels of the text (e.g., lexical, syntactic, discourse), researchers can look beyond simple measures of holistic quality (i.e., essay scores) and begin to examine and model the components of the writing process more thoroughly [27]. These models of student performance can then allow researchers and educators to provide students with more effective instruction that specifically targets their individual needs.

Broadly, NLP involves the automated calculation of linguistic text features using a computer program (or programming language) [28]. Thus, the focus of NLP primarily rests on the use of computers to understand, process, and produce natural language text for the purpose of automating certain communicative acts (e.g., providing technical support) or for studying communicative processes (e.g., examining the linguistic properties of readable texts). This technique can serve as a powerful methodological approach for researchers who are interested in examining particular aspects of the writing process [27] or for many other domains in which students produce natural language.

Researchers have employed NLP techniques within a variety of domains and contexts for the purpose of developing a better understanding the learning process [7, 24, 29, 30, 31]. For example, Varner, Jackson and colleagues (2013) used NLP tools to calculate the extent to which students' self-explanations of complex science texts contained cohesive elements [31]. Results from this study indicated that better readers produced more cohesive self-explanations than less skilled readers, indicating that automated indices of cohesion could potentially serve as a proxy for the coherence of students' mental text representations. In another study, Graesser and colleagues (2011) developed multiple components of text readability using NLP tools [29]. These components related to different dimensions of text complexity, such as narrativity, concreteness, and referential cohesion. Through the use of NLP tools, these researchers were able to develop components that provide multidimensional information about texts and the specific properties that influence students' ability to comprehend these texts successfully.

1.2.1 NLP and Writing

With regards to the writing process, NLP can serve as a particularly beneficial tool, as it can provide explicit information about students' processes and performance on the learning task. Accordingly, these NLP techniques have been used in previous research on writing, primarily with the goal of modeling human ratings of text quality [14, 30, 32]. In one particular study, Crossley and McNamara (2011) examined the linguistic indices that were significantly related to quality ratings of timed, prompt-based essays. Results of this study revealed that higher quality essays contained more sophisticated language, greater lexical diversity, more complex sentence constructions, and less frequent words. In a similar analysis, Varner and colleagues (2013) investigated differences between the linguistic indices associated with teachers' ratings of essay quality and students' self-assessments of their own essays [30]. This analysis suggested that students were less systematic in their self-assessments than teachers, at least in relation to the linguistic characteristics of the essays. Additionally, students' ratings were related to different linguistic features than the essay ratings of their teachers.

Overall, the results of these (and many other) studies suggest that NLP can serve as a powerful resource with which researchers can model the writing process at a more fine-grained size. In particular, NLP tools can potentially help researchers to develop better models of the individual differences that are important to writing proficiency (e.g., vocabulary knowledge), as well as for any other domain in which students produce natural language.

1.3 The Writing Pal

The Writing Pal (W-Pal) is an intelligent tutoring system (ITS) that was designed to provide explicit writing strategy instruction and practice to high school and early college students [18, 33] Unlike typical AWE systems, W-Pal places a strong emphasis on the instruction of writing strategies, as well as multiple forms of practice (i.e., strategy-specific practice and holistic essay writing practice).

The strategy instruction in W-Pal covers all three phases of the writing process: prewriting, drafting, and revising. Within W-Pal, these strategies are taught in individual instructional modules, which include: *Freewriting* and *Planning* (prewriting); *Introduction Building*, *Body Building*, and *Conclusion Building* (drafting); and *Paraphrasing*, *Cohesion Building*, and *Revising* (revising; see Figure 1 for a screenshot of the main W-Pal interface). Each of these instructional modules contains multiple lesson videos, which are each narrated by an animated pedagogical agent. In these videos, the agent describes and provides examples of specific strategies that are important for writing.

After viewing these lesson videos, students unlock multiple mini-games, which allow them to practice the strategies in isolation before applying them to complete essays. Within the W-Pal system, students can engage with identification mini-games, where they are asked to select the best answer to a particular question, or generative mini-games, where they produce natural language (typed) responses related to the strategy they are practicing.

One of the key features of the W-Pal system is its AWE component (i.e., the essay practice component). This system contains a word processor where students can write essays in response to a number of SAT-style prompts (teachers also have the option of adding in their own prompts to assign to students). Once a student has completed an essay, it is submitted to the W-Pal system. The W-Pal algorithm [14] then calculates a number of linguistic features related to the essay and provides summative and formative feedback to the student (see Figure 2 for a screenshot of the W-Pal feedback screen). The summative feedback in W-Pal is a holistic essay score that ranges from 1 to 6. The formative feedback in W-Pal provides information about strategies that students can employ in order to improve their essays. Once they have read the feedback, students have the option to revise their essays based on the feedback that they were assigned.

Figure 1. Main Interface of the W-Pal System

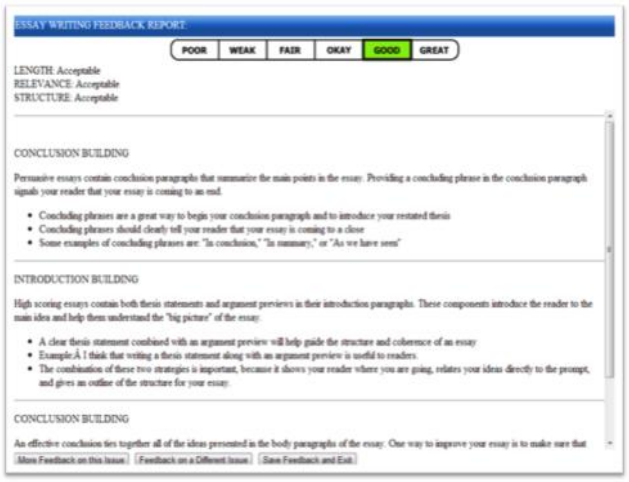


Figure 2. Example of W-Pal Feedback

2. CURRENT STUDY

The purpose of the current study is to investigate the degree to which the lexical properties of students' essays can inform stealth assessments of their vocabulary knowledge. Ideally, these assessments will serve to inform student models in the Writing Pal system and contribute to its adaptability in the form of more

sophisticated scoring algorithms, feedback, and adaptive instruction. To this end, two corpora were collected which contained essays from early college and high school students, respectively. The lexical properties of these essays were then calculated using the Tool for the Automatic Analysis of Lexical Sophistication (TAALES) [34]. TAALES is an automated text analysis tool that provides linguistic indices related to the lexical sophistication of texts. We used this tool in the current study so that we could investigate the relationships between students' vocabulary knowledge and the lexical properties of the essays. We hypothesized that these lexical indices would be significantly related to vocabulary knowledge and that they would provide reliable measures of vocabulary knowledge across two distinct student populations.

2.1 Primary Corpus

The primary corpus for this study is comprised of 108 essays written by college students from a large university campus in Southwest United States. These students were, on average, 19.75 years of age (range: 18-37 years), with the majority of students reporting a grade level of college freshman or sophomores. Of the 108 students, 52.9% were male, 53.7% were Caucasian, 22.2% were Hispanic, 10.2% were Asian, 3.7% were African-American, and 9.3 % reported other ethnicities. All students wrote a timed (25-minute), prompt-based, persuasive essay that resembled what they would see on an SAT. Students were not allowed to proceed until the entire 25 minutes had elapsed. These essays contained an average of 410.44 words (*SD* = 152.50), ranging from a minimum of 84 words to a maximum of 984 words.

2.2 Vocabulary Knowledge Assessment

Students' vocabulary knowledge was assessed using the Gates-MacGinitie (4th ed.) reading comprehension test (form S) level 10/12 [35]. This assessment is a 10-minute task, which is comprised of 45 simple sentences that each contains an underlined vocabulary word. Students were asked to read each sentence and then select the most closely related word (from a list of five choices) to the underlined word within the sentence.

2.3 Text Analyses

To assess the lexical properties of students' essays, we utilized the Tool for the Automatic Analysis of Lexical Sophistication (TAALES). TAALES is an automated text analysis tool that computes 135 indices that correspond to five primary categories of lexical sophistication: *word frequency*, *range*, *n-gram frequencies*, *academic language*, and *psycholinguistic word information* [34]. These categories are discussed in greater detail below (see 34 for more thorough information).

Word frequency indices are indicative of lexical sophistication, because high frequency words are typically learned earlier in life, are processed more quickly, and are indicative of writing quality (i.e., with high frequency words indicating lower quality writing). There are two primary forms of frequency measures: frequency bands and frequency counts. Frequency bands measure the percentage of a text that occurs in particularly frequency bands (e.g., whether they are in the most frequent 1,000 words, 2,000 words in a frequency list, etc.). Frequency counts employ reference corpora and calculate the frequency of the words in a target text within the reference corpus.

Range indices are indicative of how widely used a particular word or family of words is. Thus, unlike frequency indices, range indices do not simply calculate a raw count of a word in a particular list or corpus. Rather, range indices measure the number of individual documents that contain that word in order to determine the extent that it is used broadly. Range has been used to successfully distinguish the frequent verbs produced by L2 speakers of English from the frequent verbs produced by native English speakers [36].

N-gram frequencies emphasize units of lexical items rather than single words. In particular, n-grams consist of combinations of *n* number of words (e.g., the bigram "years ago") that frequently occur together. Bigram lists have been shown to be predictive of a speaker or writer's native language, as well as the quality of a given text.

Academic language indices measure the degree to which a text contains words that are found infrequently in natural language corpora, but frequently in academic texts. A number of academic word lists have been calculated to measure the words that are commonly used in academic texts, such as textbooks and journal articles. Thus, these indices provide a measure of how academic a text is compared to more typical texts.

Psycholinguistic word indices provide information about the specific characteristics of the words used in texts. These properties have been shown to be related to lexical decision times, lexical proficiency, and writing quality. TAALES focuses on five particular properties of words: *concreteness* (i.e., perceptions of how abstract a word is), *familiarity* (i.e., judgments of how familiar words are to adults), *imageability* (i.e., judgments of how easy it is to imagine a word), *meaningfulness* (i.e., judgments of how related a word is to other words), and *age of acquisition* (i.e., judgments of the age at which a word is typically learned).

2.4 Statistical Analyses

Statistical analyses were conducted to investigate the role of lexical properties in assessing and modeling students' vocabulary knowledge scores. Pearson correlations were first calculated between students' scores on a vocabulary knowledge measure and the lexical properties of their essays (as assessed by TAALES). The indices that demonstrated a significant correlation with vocabulary knowledge scores ($p < .05$) were retained in the analysis. Multicollinearity of these variables was then assessed among the indices ($r > .90$). When two or more indices demonstrated multicollinearity, the index that correlated most strongly with vocabulary knowledge scores was retained in the analysis. All remaining indices were finally checked to ensure that they were normally distributed.

A stepwise regression analysis was conducted to assess which of the remaining lexical indices were most predictive of vocabulary knowledge. For this regression analysis, a training and test set approach was used (67% for the training set and 33% for the test set) in order to validate the analyses and ensure that the results could be generalized to a new data set. To additionally avoid overfitting the model, we chose a ratio of 15 essays to 1 predictor, which allowed 7 indices to be entered, given that there were 108 essays included in the analysis.

A final linear regression analysis was conducted to determine the extent to which these indices could model the vocabulary knowledge of students in a different population. In particular, we investigated whether the lexical sophistication indices that were retained in the previous regression model (i.e., the regression

model for the college students) accounted for a significant amount of the variance in a second set of students' (i.e., the high school students) vocabulary knowledge.

3. RESULTS

3.1 Vocabulary Knowledge Analysis for the Primary Corpus

Pearson correlations were calculated between the TAALES indices and students' Gates-MacGinitie vocabulary knowledge scores to examine the strength of the relationships among these variables. This correlation analysis revealed that there were 45 linguistic measures that demonstrated a significant relation with vocabulary knowledge scores and did not demonstrate multicollinearity with each other. To avoid overfitting the model, we only selected the 7 indices that were most strongly correlated with vocabulary knowledge. These 7 indices are listed in Table 1 (see Kyle & Crossley for explanations of each variable) [34].

A stepwise regression analysis was calculated with these 7 TAALES indices as the predictors of students' vocabulary knowledge scores for the students in the training set. This regression yielded a significant model, F (2, 76) = 29.296, p < .001, r = .660, R^2 = .435. Two variables were significant predictors in the regression analysis and combined to account for 44% of the variance in students' vocabulary knowledge scores: mean age of acquisition log score [β =.92, t(2, 76)=6.423, p < .001] and normed count for all academic word lists [β =-.36, t(2, 76)=-2.539, p = .013]. The regression model for the training set is presented in Table 2. The test set yielded r = .600, R^2 = .360, accounting for 36% of the variance in vocabulary knowledge scores.

Table 1. Correlations between Gates-MacGinitie vocabulary knowledge scores and TAALES linguistic scores

TAALES variable	*r*	*p*
Mean age of acquisition log score	.614	<.001
Mean range (number of documents that a word occurs in) log score	-.562	<.001
Spoken bigram proportion	-.511	<.001
Mean unigram concreteness score	-.492	<.001
Mean frequency score (bigrams)	-.488	<.001
Mean frequency log score	-.476	<.001
Normed count for all academic word lists	.402	<.001

Table 2. TAALES regression analysis predicting Gates-MacGinitie vocabulary knowledge scores

Entry	Variable added	R^2	Δ R^2
Entry 1	Mean age of acquisition log score	.387	.387
Entry 2	Normed count for all academic word lists	.435	.048

The results of this regression analysis indicate that the students with higher vocabulary scores produced essays that were more lexically sophisticated. The essays contained words that were acquired at a later age, such as the words *vociferous* or *ubiquitous*, which are predicted to be learned later than words such as *toy* and *animal*. The essays also contained a greater proportion of academic words that are frequently found in academic texts, such as *financier* or *contextualized*, rather than household words such as *bread* and *house*. Hence, better writers use words that are found in academic, written language, rather than more common, mundane language. Notably, these two indices, age of acquisition, and academic words, are likely to correlate with indices related to the frequency or familiarity of words in language. However, in this case, they more successfully captured students' vocabulary knowledge from their writing samples compared to simple frequency or familiarity indices.

3.2 Generalization to a New Data Set

Our second analysis specifically tested the ability of the linguistic indices to predict the Gates-MacGinitie vocabulary knowledge scores of students in a completely separate population. To address this question, we collected a test corpus of essays written by high school students and analyzed the lexical properties of these essays. Specifically, we calculated the *mean age of acquisition log score* and the *normed count for all academic word lists*, as these were the two indices retained in the previous regression model. These indices were then used as predictors in a regression model to predict students' vocabulary knowledge.

3.3 Test Corpus

The test corpus in this paper was collected as part of a larger study (n = 86), which compared the complete Writing Pal system to the AWE component of the system. Here, we focus on the pretest essays produced by these participants. All participants were high-school students recruited from an urban environment located in the southwestern United States. These students were, on average, 16.4 years of age, with a mean reported grade level of 10.5. Of the 45 students, 66.7% were female and 31.1% were male. Students self-reported ethnicity breakdown was 62.2% were Hispanic, 13.3% were Asian, 6.7% were Caucasian, 6.7% were African-American, and 11.1% reported other. All students wrote a timed (25-minute), prompt-based, argumentative essay that resembled what they would see on the SAT. Students were not allowed to proceed until the entire 25 minutes had elapsed. These essays contained an average of 340.84 words (SD = 124.31), ranging from a minimum of 77 words to a maximum of 724 words. Finally, these students completed the same vocabulary knowledge assessment as the students in the previous corpus.

3.4 Vocabulary Knowledge Analysis for the Test Corpus

The two TAALES indices (i.e., *mean age of acquisition log score* and the *normed count for all academic word lists*) were entered as predictors of students' Gates-MacGinitie vocabulary knowledge scores. This regression yielded a significant model, F (2, 83) = 8.521, p < .001, r = .413, R^2 = .170. Only one of the variables was a significant predictor in the regression analysis: mean age of acquisition log score [β =.54, t(2, 83)=3.666, p < .001]. This model suggests that the regression model generated with the primary corpus partially generalized to a new data set. One of the indices accounted for a significant amount of the variance in students' vocabulary knowledge scores. However, this variance was smaller than the variance accounted for in the primary corpus.

4. DISCUSSION

Computer-based writing systems provide students with learning environments in which they can receive writing instruction and engage in deliberate practice [10]. One of the major difficulties that developers of these systems face, however, is the ability to provide instruction and feedback that is *personalized* to individual student users. Developers of these systems often rely on NLP techniques to assess the quality of individual essays; however, it has been relatively unclear whether these NLP techniques can be used to assess relevant individual differences among students.

In the current study, we used NLP techniques to develop stealth assessments of students' vocabulary knowledge. Vocabulary knowledge is an important component of the writing process [5, 19]; thus, our aim was to determine whether we could assess and model individual differences in this knowledge by calculating the lexical sophistication of students' essays. Specifically, an automated text analysis tool was used to analyze the lexical properties of the essays. This tool (TAALES) provided information about the lexical sophistication of the essays at multiple levels (e.g., *word frequency*, *range*, *n-gram frequencies*, *academic language*, and *psycholinguistic word information*). The results revealed that these indices were able to significantly model students' vocabulary knowledge scores. Additionally, these findings were able to predict students' vocabulary scores on a separate data set.

The TAALES correlation analysis revealed that there were 45 lexical sophistication indices that significantly correlated with students' vocabulary knowledge. This is important, because it indicates that individual differences in students' vocabulary knowledge could be detected by analyzing the lexical items that students used in their essays. Further, the regression analyses revealed that the *psycholinguistic word information* and *academic language* indices provided the most predictive power in the model (as opposed to simple measures of word frequency or familiarity), with indices of age of acquisition and academic words accounting for 44% of the variance in the vocabulary scores. Thus, students with greater vocabulary knowledge tended to produce essays with words that are judged to be acquired later in life and were more academic in nature.

Importantly, the follow-up regression analysis revealed that these two TAALES indices accounted for a significant amount of the variance in vocabulary scores for a separate corpus of student essays. In particular, the age of acquisition variable was able to account for approximately 17% of the variance in students' vocabulary knowledge scores. This finding provides confirmation that the automated lexical sophistication indices could be used across two separate data sets to model vocabulary knowledge.

It is important to note, however, that this variable accounted for a significantly smaller amount of the variance in this test corpus than in our primary corpus. This suggests that individual differences may manifest in the properties of students' essays in different ways depending on the specific context. For instance, in this study, the students who produced essays for the two corpora were in college and high school, respectively. Thus, variations in vocabulary knowledge might have influenced the high school and college students' writing process differentially based on the other knowledge, skills or strategies that they had available to them. The results of this follow-up analysis suggest, therefore, that computer-based learning environments may need to rely on separate models for students from different populations. Although the same techniques may be able to be used for all student groups (e.g., the use of NLP), the specific indices in the models may need to be modified across different populations.

Overall, the results from the current study suggest that NLP indices can be utilized to develop stealth assessments of students' skills. When taken together, two indices of lexical sophistication accounted for nearly half of the variance in students' vocabulary knowledge scores. These findings are important, because they indicate that students' individual differences can manifest in the ways that they produce essays. Thus, linguistic analyses of essays (and any other natural language input) may provide useful information about individual students' knowledge and skills. Here, we only analyzed students' vocabulary knowledge at pretest (i.e., before they received any training or feedback). In the future, additional studies will be conducted to specifically examine how these stealth assessments of vocabulary knowledge will change throughout training and how they will serve to inform consistently updating student models.

An additional area for future research lies in the assessment of other individual difference variables. In the current study, we solely analyzed the lexical properties of students' essays because we were focusing on one particular individual difference measure: vocabulary knowledge. In future studies, however, it will be important to consider additional linguistic indices that may be related to other specific constructs of interest. For instance, if we aim to model students' attitudes during writing practice, lexical sophistication indices may provide little valuable information. Instead, we may turn to measures of semantic information, such as the tone or themes found in the essays. Similarly, if we are assessing students' reading comprehension skills, it may be more fruitful to include cohesion indices, which describe the degree to which information in a text is explicitly connected.

In conclusion, the current study utilized the NLP tool, TAALES, to investigate the efficacy of NLP techniques to inform stealth assessments of vocabulary knowledge. Eventually, we expect that this stealth assessment will enhance our student models within the W-Pal system and allow us to provide students with more pointed feedback and instruction. More broadly, the current study suggests that NLP techniques can (and should) be used to help researchers and system developers build stealth assessments and student models in computer-based learning environments. These models can ultimately be used to provide more personalized and adaptive computer-based instruction for students.

While a wealth of studies awaits to answer myriad questions on *how* to construct the most powerful models of individual differences without having to administer the tests, this is a strong step forward in demonstrating the feasibility of such stealth measures.

5. ACKNOWLEDGEMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080589 to Arizona State University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

6. REFERENCES

[1] National Commission on Writing. 2003. *The Neglected "R."* College Entrance Examination Board, New York.

[2] Baer, J. D., and McGrath, D. 2007. The reading literacy of U.S. fourth-grade students in an international context: Results from the 2001 and 2006 Progress in International Literacy Study (PIRLS). National Center for Educational Statistics, Institute of Education Sciences, U.S. Department of Education.

[3] National Assessment of Educational Progress. 2009. The Nation's Report Card: Writing 2009. Retrieved Nov. 5, 2012, nces.ed.gov/nationsreportcard/writing.

[4] National Assessment of Educational Progress. 2011. The Nation's Report Card: Writing 2011. Retrieved Nov. 5, 2012, nces.ed.gov/nationsreportcard/writing.

[5] Allen, L. K., Snow, E. L., Crossley, S. A., Jackson, G. T., and McNamara, D. S. 2014. Reading comprehension components and their relation to the writing process. *L'année psychologique/Topics in Cognitive Psychology*, 114, (2014) 663-691.

[6] Flower, L. and Hayes, J. 1981. Identifying the organization of writing processes. In L. Gregg and E. Steinberg (Eds.), Cognitive processes in writing. Erlbaum & Associates, Hillsdale, NJ, 3-30.

[7] Allen, L. K., Snow, E.L., and McNamara, D. S. 2014. The long and winding road: Investigating the differential writing patterns of high and low skilled writers. In J. Stamper, S. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (London, UK, July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, 304-307.

[8] Johnstone, K.M., Ashbaugh, H., and Warfield, T.D. 2002. Effects of repeated practice and contextual writing experiences on college students' writing skills. *Journal of Educational Psychology* (2002), 94, 305–315.

[9] Kellogg, R., and Raulerson, B. 2007. Improving the writing skills of college students. *Psychonomic Bulletin and Review*, 14, (2007), 237-242.

[10] Allen, L. K., Jacovina, M. E., and McNamara, D. S. in press. Computer-based writing instruction. In C. A. MacArthur, S. Graham, and J. Fitzgerald (Eds.), *Handbook of Writing Research.*

[11] Dikli, S. 2006. An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment*, (2006), 5.

[12] Deane, P. 2013. On the relation between automated essay scoring and modern views of the writing construct. *Assessing Writing*, 18, (2013), 7-24.

[13] Shermis, M., and Burstein, J. (Eds.). 2003. *Automated essay scoring: A cross-disciplinary perspective*. Erlbaum, Mahwah, NJ.

[14] McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., and Dai, J. 2015. Hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23, (2015), 35-59.

[15] Warschauer, M., & Ware, P. 2006. Automated writing evaluation: defining the classroom research agenda. *Language Teaching Research, 10*, (2006), 1-24.

[16] Attali, Y., and Burstein, J. 2006. Automated essay scoring with e-rater V.2. *Journal of Technology, Learning, and Assessment*, 4, (2006), 3.

[17] Crossley, S. A., Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Using automated indices of cohesion to evaluate an intelligent tutoring system and an automated writing evaluation system. In K. Yacef et al (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED).* Springer, Heidelberg, Berlin, 269-278.

[18] Roscoe, R. D., Varner, L. K., Weston, J. L., Crossley, S. A., and McNamara, D. S. 2014. The Writing Pal intelligent tutoring system: Usability testing and development. *Computers and Composition*, 34 (2014), 39-59.

[19] Crossley, S. A., Allen, L. K., Snow, E. L., & McNamara, D. S. in press. Pssst…textual Features… there is more to automatic essay scoring than just you! In *Proceedings of the 5th International Learning Analytics and Knowledge Conference (LAK)*, Poughkeepsie, NY.

[20] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. In S. Tobias & J. D. Fletcher (Eds.), Computer Games and Instruction. Information Age Publishers, Charlotte, NC, 503-524.

[21] Shute, V. J., and Kim, Y. J. 2013. Formative and stealth assessment. In J. M. Spector, M. D. Merrill, J. Elen, and M. J. Bishop (Eds.), *Handbook of Research on Educational Communications and Technology (4th Edition)*. Lawrence Erlbaum Associates, Taylor & Francis Group, New York, NY, 311-323.

[22] Shute, V. J., Ventura, M., Bauer, M. I., and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. Cody & P. Vorderer (Eds.), Serious games: Mechanisms and effects. Routledge, Mahwah, NJ, 295-321.

[23] Snow, E. L., Jacovina, M. E., Allen, L. K., Dai. J., and McNamara, D. S. 2014. Entropy: A stealth assessment of agency in learning environments. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining*, (London, UK, July 4 -7, 2014),Springer Berlin Heidelberg, 241-244.

[24] Allen, L. K., Snow, E. L., and McNamara, D. S. in press. Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques. In *Proceedings of the 5th International Learning Analytics and Knowledge Conference (LAK)*, Poughkeepsie, NY.

[25] Brusilovsky, P. 1994. The construction and application of student models in intelligent tutoring systems. *Journal of Computer and Systems Science International*, 23, (1994), 70-89.

[26] Vanlehn, K. 2006. The behavior of tutoring systems. International journal of artificial intelligence in education, 16 (2006), 227-265.

[27] Crossley, S. A., Allen, L. K., Kyle, K., and McNamara, D. S. 2014. Analyzing discourse processing using a simple natural

language processing tool (SiNLP). *Discourse Processes*, 51, 511-534.

[28] Crossley, S. A. 2013. Advancing research in second language writing through computational tools and machine learning techniques: A research agenda. *Language Teaching, 46* (2013), 256-271.

[29] Graesser, A. C., McNamara, D. S., and Kulikowich, J. M. 2011. Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher, 40*, (2011), 223-234

[30] Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Evaluative misalignment of 10^{th}-grade student and teacher criteria for essay quality: An automated textual analysis. *Journal of Writing Research, 5*, (2013), 35-59.

[31] Varner, L. K., Jackson, G. T., Snow, E. L., & McNamara, D. S. (2013). Does size matter? Investigating user input at a larger bandwidth. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), Proceedings of the 26th Annual Flordia Artificial Intelligence Research Society (FLAIRS) Conference (pp. 546-549). Menlo Park, CA: The AAAI Press.

[32] Crossley, S. A., and McNamara, D. S. Text coherence and judgments of essay quality: Models of quality and coherence. In L. Carlson, C. Hoelscher, & T.F. Shipley (Eds.), Proceedings of the 33rd Annual Conference of the Cognitive Science Society. (pp. 1236-1231). Austin, TX: Cognitive Science Society.

[33] Roscoe, R. D., and McNamara, D. S. 2013. Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105, (2013), 1010-1025.

[34] Kyle, K. and Crossley, S. A. in press. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly* (in press).

[35] MacGinitie, W.H., MacGinitie, R.K., Maria, K., and Dreyer, L.G.: Gates-MacGinitie Reading Test (4th ed.). The Riverside Publishing Company, Itasca, 2000.

[36] Crossley, S. A., Cobb, T., and McNamara, D. S. 2013. Comparing count-based and band-based indices of word frequency: Implications for active vocabulary research and pedagogical applications. *System*, 41, (2013), 965-981.

Automatic Identification of Nutritious Contexts for Learning Vocabulary Words

Jack Mostow, Donna Gates, Ross Ellison, Rahul Goutam
Project LISTEN (www.cs.cmu.edu/~listen), School of Computer Science, Carnegie Mellon University
RI-NSH 4103, 5000 Forbes Avenue, Pittsburgh, PA 15213-3980, USA
011 (412) 268-1330
mostow@cmu.edu, dmg@alumni.cmu.edu, rpelliso@andrew.cmu.edu, rgoutam@cmu.edu

ABSTRACT

Vocabulary knowledge is crucial to literacy development and academic success. Previous research has shown learning the meaning of a word requires encountering it in diverse informative contexts. In this work, we try to identify "nutritious" contexts for a word – contexts that help students build a rich mental representation of the word's meaning. Using crowdsourced ratings of vocabulary contexts retrieved from the web, AVER learns models to score unseen contexts for unseen words. We specify the features used in the models, measure their individual informativeness, evaluate AVER's cross-validated accuracy in scoring contexts for unseen words, and compare its agreement with the human ratings against the humans' agreement with each other. The automated scores are not good enough to replace human ratings, but should reduce human effort by identifying contexts likely to be worth rating by hand, subject to a tradeoff between the number of contexts inspected by hand, and how many of them a human judge will consider nutritious.

Keywords

Vocabulary learning, crowdsourcing, automated scoring, regression models.

1. INTRODUCTION

Years of research on vocabulary learning have found that vocabulary is a bottleneck to comprehension [1], shown that vocabulary instruction benefits students' word learning and text comprehension [2-5], and identified several principles of effective vocabulary instruction [6-12]. The principle relevant here is that vocabulary learning requires exposure to diverse informative example contexts in order to develop a rich mental representations of word meanings and their relations to other words.

This paper describes AVER ("Automatic Vocabulary Example Rater"), an attempt to automatically identify "nutritious" contexts – example uses of a word that should help in learning its meaning. (***Aver*** is itself a vocabulary word that means ***assert***.) This work is part of a larger project that supplied our training and test data in the form of target vocabulary words, example contexts in which they occur, and human ratings of their nutritiousness. The contexts were retrieved from the web by DictionarySquared.com, an online high school vocabulary tutor that searches the web for a given target word in order to find candidate contexts that contain it. DictionarySquared aims to pick contexts a few dozen words long, preferring to start and end at boundaries between sentences, paragraphs, or HTML blocks.

This paper describes how AVER trains and evaluate models to predict the nutritiousness of such contexts, based on human ratings crowdsourced using Amazon Mechanical Turk.

Ideally AVER would identify a set of examples that maximizes the amount of actual student learning from a given number of contexts, taking into account the diversity of multiple contexts for the same word, and possibly even their relation to example contexts for other target vocabulary words to learn. However, this paper focuses on the initial problem of predicting the suitability of individual contexts, using crowdsourced human estimates instead of students' subjective ratings of contexts, or objective measures of their actual learning gains.

1.1 Relation to Prior Work

Some previous work has addressed the problem of finding suitable example contexts to support vocabulary learning, but differed in one or more respects from the work reported here. REAP [13] selected examples from an already-vetted corpus, based on specified selection criteria such as student interests. VEGEMATIC [14] constructed 9-word contexts centered on a given target vocabulary word by concatenating overlapping 5-grams from the Google *n*-gram corpus, based on heuristic constraints and preferences; only some of them were good enough to use, but hand-vetting them was faster than composing good examples by hand. Follow-on work [15] extended VEGEMATIC to generate contexts for a particular sense of a target word. AVER also seeks to identify example contexts suitable for vocabulary learning, but addresses a different goal than both these projects: instead of applying explicit hand-crafted heuristics, AVER learns to predict crowdsourced ratings by human judges.

The rest of the paper is organized as follows. First we describe our data set. Then we describe the features we used, tried but dropped, or identified but didn't implement. Next we describe and evaluate how AVER rates contexts. Finally we conclude.

2. DATA SET

The data for this work consists of a vocabulary word and a context that contains at least one instance of the vocabulary word and that illustrates usage of the vocabulary word. The overall data set includes 75,844 contexts for 1,000 vocabulary words, comprising 100 words from each of 10 difficulty bands based on their Standardized Frequency Index [16], a measure of log frequency in a text corpus, adjusted by dispersion across multiple domains.

Dr. Margaret G. McKeown, an international expert on vocabulary learning and instruction, rated 93 contexts based on three criteria – the typicality of the usage of the vocabulary word in the context, the degree to which the context constrains the meaning of the vocabulary word, and the comprehensibility of the context for students. Thus the expert provided three ratings of each context, one on each criterion, ranging from 1 (very poor) to 5 (very good). These data helped in developing a rating scale. However, it would have been infeasible to obtain expert ratings of enough contexts to train good models.

Therefore, using Amazon Mechanical Turk, 13,270 contexts were each rated by 10 amateur raters who passed a brief test of their performance on this task: "Based on context, rate how helpful the text is for helping a high school student understand the meaning of the target word. A helpful context is one that reinforces a word's meaning and is understandable to high school students." Contexts ranged in length from 18 to 137 words, with median 63.

Raters differed in how many contexts they rated, ranging from several to hundreds. They rated contexts on a 5-point scale:

- 4 = Very Helpful: After reading the context, a student will have a very good idea of what this word means.
- 3 = Somewhat Helpful
- 2 = Neutral: The context neither helps nor hinders a student's understanding of the word's meaning.
- 1 = Bad: The context is misleading or too difficult.
- 0 = Otherwise inappropriate for high school students.

We used the mean of their 10 ratings to label our training and testing data. Inter-rater standard deviation averaged 0.81, so standard error averaged 0.27. We labeled the 4107 contexts with mean rating at or above 3 as "good," and the 9150 contexts with mean rating below 3 as "bad."

3. FEATURES USED

The remaining 62,574 contexts were not rated by humans. To rate their nutritiousness automatically, AVER uses the human-labeled data to train and test regression models to predict the ratings of unseen contexts for unseen words, or to predict the probability that a context is "good," i.e., its rating is greater than or equal to 3.

To train these models, we extract features of the vocabulary word and context we consider likely to be informative in predicting its human rating. We normalize every feature as a z-score by subtracting the mean value for that feature and dividing by its standard deviation. By translating all feature values onto a common scale, normalization makes their regression coefficients comparable. Normalization does not affect a feature's correlation with Turker ratings or other features because correlation is invariant under constant addition or multiplication. We assign a z-score of zero to features with undefined values, so that they have no impact on model output.

To describe various types of features, illustrate their values, explain their meaning, and discuss the intuition underlying them, we will use the following example context for the vocabulary word ***alleviate***, with mean Turker rating = 3.7, i.e. quite good:

> ***It is ironic that students are pressured to do well in school in order to continue participating in extracurricular activities, yet these after school activities are just what they need to relieve stress. Sports clubs and even being involved in student government can help alleviate stress. They allow us to get away from school pressure and enjoy ourselves.***

3.1 Comprehensibility

Our goal is to help students learn the typical usage of a vocabulary word by providing them with example contexts. If the example contexts are too difficult to understand, they will not be very helpful to students. Thus indicators of comprehensibility are useful features in predicting the rating of a context.

Rarer words are typically harder. The log frequency of ***alleviate***, i.e., the log of its unigram count (1,596,620) divided by the total number of tokens (1,024,908,267,229) in the Google *n*-grams corpus, is –13.4 (z-score = –0.090), placing it in the third most common of 10 word bands (z-score = 0.150). This feature of the target word is the same for all its contexts, but helps control for target word frequency in general models to predict context ratings.

The more and longer the words in a context, the harder it is to understand. The example context has 58 words (z-score = –0.235, which on average are 5.1 letters long (z-score = 0.358), not counting spaces or punctuation.

Flesch-Kincaid scores for reading ease and grade level are widely used to assess readability, and we compute them for contexts:

Reading ease =

$$\mathbf{206.835 - 1.015 \times \frac{total\ words}{total\ sentences} - 84.6 \times \frac{total\ syllables}{total\ words}}$$

Grade level =

$$\mathbf{0.39 \times \frac{total\ words}{total\ sentences} + 11.8 \times \frac{total\ syllables}{total\ words} - 15.59}$$

A higher reading ease score characterizes text as easier to read and understand. The reading ease score ranges from 0 to 100. The reading ease score for our example context is 47.18, indicating that it is moderately difficult (z-score = –0.015). Flesch-Kincaid scores depend on how syllables, words, and sentences are counted, and hence differ from one implementation to another, but not by much. Thus Microsoft Word reports a reading ease of 48.6 for this paragraph.

A higher grade level score indicates a context that is more difficult to read and understand. The grade level roughly translates to the number of years of education required to understand the context. The grade level score for our example context is 11.48 (z-score = –0.217), compared to 11.2 in Microsoft Word.

Mean human ratings correlated 0.009 with log of target frequency, 0.023 with word band, –0.082 with context length, –0.039 with average word length, 0.043 with reading ease, and –0.030 with grade level.

3.2 Local Predictability

AVER extracts local predictability features from a 9-word context centered on the target word (e.g. ***student government can help alleviate stress . They allow***). They estimate the probability of the target word given a local context containing the target word. Five of these local contexts are 5 words long, four are 4 words long, three are 3 words long, two are 2 words long, and the target itself can be considered a 1-word context, so there are 15 probabilities. The submitted version of this paper used all 15 of these probabilities as features.

To estimate these probabilities, AVER uses the Google *n*-grams tables [16] based on over a trillion words from the web. These tables specify the frequency of every word unigram, bigram, trigram, 4-gram, and 5-gram with at least 40 occurrences. Thus AVER can use them to estimate such conditional probabilities up to a context length of 5 words. For example, it would estimate the conditional probability of ***alleviate*** given the 5-word local context ***government can help ____ stress*** as a fraction whose numerator is the frequency of the 5-gram ***government can help alleviate stress*** and whose denominator is the summed counts of all 5-grams of the form ***government can help * stress.***

AVER log-transforms the probability estimates to reduce their enormous dynamic range, and normalizes the log probabilities as z-scores, which it uses as features to measure local predictability.

If the numerator is zero, AVER smoothes it to 1. The numerator is zero for 88% to 93% of the 5-word contexts, varying by the position of the target word. E.g., ***help alleviate stress . They*** is not in the 5-gram table. The numerator is zero for 68% to 78% of the 4-word contexts, 33% to 44% of the 3-word contexts, and 8% to 9% of the 2-word contexts.

What if the denominator is zero (e.g. no 5-grams of the form ***government can help * stress*** are listed in the 5-gram table)? The denominator is zero for 82% to 86% of our 5-word contexts that contain the target word; the percentage varies by its position in the context. Likewise, the denominator is zero for 47% to 57% of the 4-word contexts, and 33% to 44% of the 3-word contexts.

In the submitted version of this paper, we translated the resulting undefined probability into a z-score of zero, so that it would neither increase nor decrease the output of our predictive models. However, the effect was that some features, especially for 5-grams, were mostly zero in the training data. Could we do better?

Inspired by a reviewer comment, we implemented a new version, called AVER.b (b for "backoff") based on an idea from statistical language modeling: in the absence of data about a particular *n*-gram, back off to successively shorter *n*-grams. For instance, if the denominator is zero because no 5-grams of the form ***government can help * stress*** are in the 5-gram table, AVER.b looks for 4-grams of the form ***government can help ** or ***can help * stress.*** If AVER.b finds both, it backs off to whichever yields a higher probability for the target word, on the assumption that it is more informative. If it finds neither, it backs off to trigrams, then bigrams, then finally the unigram ***alleviate***.

For our example, 5-word contexts of the form ***can help * stress .*** are the only ones listed in the 5-gram table, with frequency 109 for ***alleviate***, 455 for ***reduce***, 329 for ***relieve***, and 49 for ***with***. The numerator 109 and denominator 942 yield log probability –2.16.

For the other 4 positions, AVER.b backs off to 4-grams. Its 4-gram table yields non-zero denominators for 4-word contexts of the form ***help * stress .*** (4829), ***can help * stress*** (6484), and ***government can help ** (6765). It yields non-zero numerators for ***help alleviate stress .*** (330) and ***can help alleviate stress*** (325) but zero for ***government can help alleviate***, which it smoothes to 1, yielding respective log probabilities of –2.68, –2.99, and –8.82.

AVER.b finds no 4-grams of the form ***** stress . They***, so it backs off to 3-grams, using the count of ***alleviate stress .*** (2120) as numerator and the number of 3-grams of the form ***** stress .*** (1599767) as denominator, yielding log probability -6.63.

To speed up such computations, we had years earlier indexed each table by various sequences of *n*-gram positions designed to quickly retrieve all rows matching the values specified for any subset of positions. Table 1 lists these indexes, which took weeks of computer time to build because the tables have so many rows.

Table 1: Indexes constructed for Google *n*-grams tables

Table:	# rows:	Indexed by:
unigram	13,588,391	1, frequency
bigram	314,843,401	12, 21
trigram	977,069,902	123, 312, 23
4-gram	1,313,818,354	1234, 234, 314, 412, 24, 34
5-gram	1,176,470,663	12345, 5432, 3145, 2541, 1523, 432

For instance, to look up the count of the 5-gram ***government can help alleviate stress*** efficiently, both versions of AVER use the index 12345. This count is the numerator for estimating the probability of ***alleviate*** at word 4 given a 5-word context. To find all 5-grams of the form ***government can help * stress***, AVER uses the index 1523. If it finds any, it sums their frequencies as the denominator. If not, AVER.b backs off as described above. It then uses the index 1234 to look up the 4-grams ***government can help alleviate*** and ***can help alleviate stress*** as well as 4-grams of the form ***government can help ****. AVER uses the index 412 to find 4-grams of the form ***can help * stress***.

This method if necessary estimates the conditional probability of ***alleviate*** given the local bigram context ***help*** ____ as the bigram frequency of ***help alleviate*** divided by the summed frequency of all bigrams of the form ***help ****. However, there are 28,578 bigrams of this form, and it takes non-trivial time to retrieve them in order to compute their summed frequency of 270,480,813. Instead, both versions of AVER would approximate this sum as the unigram frequency of ***help***, namely 271,840,666, which it can retrieve quickly from a single row of the Google unigram table. This over-estimate includes all bigrams of the form ***help **** that occurred fewer than 40 times in the Google *n*-grams corpus and hence do not appear in the Google bigrams table. This approximation is possible only if the blank falls at the start or end of the *n*-gram. Thus it can approximate the number of trigrams of the form ***can help **** or ***** stress .***, but not ***help * stress***. The approximation was not necessary for 4- or 5-grams because they typically have many fewer rows in the *n*-gram table.

A target word can occur at *n* different positions in a word window of size *n*, with a separate probability for each window size and position within the window, represented as a log probability. Consequently, original AVER's local predictability features consist of 1 + 2 + 3 + 4 + 5 = 15 different log probabilities. For our example context, their respective z-scores are –0.090; –0.120, 0.740; 0.431, 1.340, –6.775; 0, 0.972, 0.909, –0.351; and 0, 0, 0.603, 0, 0. The z-scores of zero reflect the sparsity of *n*-grams as *n* increases.

The relative weights of these 15 z-scores reflect the overall local predictability of the target word ***alleviate*** in the local context ***student government can help alleviate stress . They allow.*** AVER sets these weights empirically as part of optimizing the weights for all our features, not just these 15. Correlations of the 15 features with human ratings range from 0.138 for logP(target w1 | ___ w1) to –0.009 for logP(target w1 w2 w3 w4 | ___ w1 w2 w3 w4). I.e., before ***stress***, ***alleviate*** is likelier to occur, but before ***stress . They allow***, the word ***alleviate*** is a bit less likely to occur.

In contrast, AVER.b uses just five local predictability features, one for each position in a 5-word context. In our example, their respective z-scores are 0.071, 1.006, 1.157, 0.944, and –0.457. The third value is largest, i.e. ***can help __ stress .*** is the 5-word context that most strongly predicts ***alleviate***. The five features correlate with mean Turker ratings at 0.055, 0.038, 0.065, 0.042, and 0.062.

To estimate the probability of the target word at word *i* given a 5-word window, AVER.b uses *n*-grams whose length n_i varies by the amount of backoff. To reflect the relative specificity of the evidence for each estimated probability, we tried weighting it by

$$n_i / \sum_{i=1}^{5} n_i$$

but it made model fit slightly worse, so we decided not to weight by *n*-gram length. Perhaps weighting it differently would help.

3.3 Topicality

Topicality features measure relatedness of the target vocabulary word to other content words in the context. The intuition behind using such features is that a context containing a typical usage of the target vocabulary word is likely to contain other content words that co-occur frequently with the target vocabulary word or are distributionally similar to it, i.e. tend to co-occur with the same words that the target word co-occurs with. The DISCO tool [17] at www.linguatools.de measures the co-occurrence of two words within 3 words of each other ("S1") and their distributional similarity ("S2") in a specified corpus, such as the British National Corpus (BNC), which contains 119 million tokens and 122,000 unique content words in "samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English, both spoken and written, from the late twentieth century" [18]. AVER uses DISCO to compute co-occurrence and distributional similarity between the target vocabulary word and each content word in the context.

To score the overall topicality of a context for the target word, we must aggregate the relatedness scores for the individual context words. Typically only a few of the context words are strongly related to the target word. Consequently, the overall average relatedness of the context dilutes their influence. Instead, AVER averages relatedness over just the most related ***k*** words of the context. In informal tests of different values of ***k***, the average of the top 5 relatedness scores did best at predicting human ratings.

Thus AVER computes two topicality scores for a context. The co-occurrence z-score for our example context is 5.063. Context words that tend to co-occur with the target vocabulary word ***'alleviate'*** include ***'pressure'*** and ***'stress'.*** The distributional similarity z-score for our example context is 1.497. The context word with the highest distributional similarity to ***'alleviate'*** is ***'relieve'.*** DISCO's S1 and S2 scores based on BNC correlated with mean human context ratings at 0.060 and 0.025, respectively.

4. FEATURES TRIED BUT ABANDONED

We now discuss several features that we experimented with but do not use in AVER, either because they hurt predictive accuracy in informal small experiments, or because they were too complex to compute efficiently.

4.1 Topicality Based on Google *N*-grams

As explained above, AVER computes context typicality using DISCO co-occurrence and similarity scores based on the British National Corpus. These scores suffer from data sparsity in the case of less-frequent words. In contrast, the Google *n*-grams corpus is based on over 10,000 times as much text, namely a trillion words of Web text. Not only is this corpus four orders of magnitude larger than BNC, it is also more relevant to the example contexts because they too consist of Web text.

Although the Google *n*-grams corpus is already in the form of *n*-grams rather than the text they are based on, its size makes it computationally expensive to compute similarity scores from it, so in previous work we had precomputed and indexed a table of the number of *n*-grams containing a given pair of words at a distance of 1, 2, 3, or 4 words, and those *n*-grams' summed frequency. However, this table has 921,643,327 rows. Despite efficient indexing, a target word's co-occurrences take considerable time to look up – over 30 seconds for ***alleviate***. To compute distributional similarity with reasonable speed, we therefore estimated it from the first few hundred rows. Unfortunately, the resulting feature harmed rather than helped model accuracy. To compute more predictive estimates of co-occurrence and distributional similarity based on Google *n*-grams, it might help to sample them more judiciously, and to adjust better for differences among target words to make estimates comparable.

4.2 Language Model Probability

To quantify the likelihood of a given context occurring in English, we used a language model trained on English text using the NLTK language model package at www.nltk.org. The motivation for this feature was to penalize contexts that contain ill-formed or incomplete sentences. We dropped this feature because it did not improve predictive accuracy, but maybe other variants of it might.

4.3 Weighted Human Ratings

Apart from different features that we tried out but did not include in the final model, we also investigated methods to improve the accuracy of the labels computed by averaging 10 raters' ratings of each context. These methods weighted the average based on each rater's degree of agreement with expert ratings of other contexts. The more closely the rater agreed with the expert on the contexts they both rated, the more accurately we expected the rater to rate contexts that the expert did not rate.

However, most raters did not overlap with the expert in terms of which contexts they rated. We therefore extended the method transitively to rate such raters based on their degree of agreement with raters who had no***n***-zero overlap with the expert, and on how closely those raters agreed with the expert on the contexts they both rated.

We also used the overlapping contexts to train a model to predict a rater's ***expected*** degree of agreement with the expert, based on features of the rater such as the total number of contexts he or she had rated. We hoped to use this model to predict agreement with the expert even for raters with zero overlap. However, the expert rated only 93 contexts, so very few raters overlapped with the expert. Even they overlapped too little to accurately estimate the rater's agreement with the expert. We therefore abandoned the approach of rating raters by their actual or expected agreement with the expert, and using it to weight the individual ratings averaged to rate a given context. Rating raters might be effective given a larger sample of expert ratings, and greater overlap of the raters with the expert.

5. FEATURES FOUND BUT NOT USED

Based on expert linguistic analysis of over 200 contexts whose human and automated ratings differed drastically, we identified

some syntactic and semantic features not exploited by the current models, and likely to improve them.

5.1 Syntactic Features

Additional syntactic features of a context could be computed by parsing it with the Stanford parser, and extracting them from the parse tree with Tsurgeon and Tregex, using the tools at nlp.stanford.edu/software/corenlp.shtml [19]. commondatastorage.googleapis.com/books/syntactic-ngrams/index.html [20] is a corpus of syntactic *n*-grams that provides counts of dependency tree fragments, which could be used to rate the plausibility of the parse and to infer likely dependency relations among context words. If for some reason part-of-speech tagging the context is feasible but parsing it is not, its dependency relations could be inferred from its part-of-speech *n*-grams [21].

Informative syntactic features include the direct object of a target verb, e.g. ***abdicate*** in ***Edward abdicated the throne***, and the objects of prepositions following a target word, e.g. ***keen*** in ***They are very keen on education***. Another syntactic feature comes from coordinate constructions, e.g., ***it is characterized by inconsistency and vagary***. The coordinated conjuncts are likely to be semantically similar or even synonymous.

It might also be useful to incorporate syntactic information into the current *n*-gram features. In particular, disaggregating *n*-gram features by the target word's part of speech in the context would exploit systematic statistical differences between parts of speech. For instance, if the target word is a verb, its subject is likely to precede it, and shed semantic light on what sorts of agents can perform the verb. Conversely, if the target word is an adjective, the noun phrase after it illustrates what the adjective can modify.

5.2 Semantic Features

Our analysis of misrated contexts found that spuriously low similarity ratings are often caused by lack of co-occurrences due to sparse data for less-frequent words. This deficiency might be addressed by augmenting BNC data with definitions, Wordnet gloss examples, and Google *n*-grams, provided the computational issues discussed earlier are satisfactorily addressed. For example, if we use Google *n*-gram features only where BNC data is too sparse, they might not pose such computational bottlenecks. Likewise, we could complement DISCO metrics of semantic similarity with features based on WordNet links from a target word to any of its synonyms, antonyms, hypernyms, and hyponyms that occur in the context.

6. AUTOMATED RATING OF CONTEXTS

AVER and AVER.b use the features described above in two types of models to rate contexts automatically for a given target word. The linear regression model predicts the mean human rating of a context. The logistic regression model is a binary classifier: it predicts whether a context is "good" (rated 3 or above) or "bad" (below 3).

We could run these models on all 75,844 contexts, but we can evaluate the models only on the 13,270 contexts rated by humans. To estimate the performance of both models on unseen data, we therefore use 5-fold cross-validation: We split the target words randomly into 5 equal subsets so as to partition the contexts into 5 subsets ("folds") with no overlap in target words between folds. For each fold we train both models on the other 4 folds, measure their performance on the held-out fold, and average over the held-out folds to estimate predictive accuracy on unseen target words – including the 62,574 unrated contexts, assuming they're similar.

To estimate performance fairly on unseen target words, it is essential to avoid overlap in target words between folds. Otherwise even if contexts do not overlap across folds, overlap in target words causes overfitting and inflates estimated performance on unseen data, especially if the training and test sets contain very similar contexts. Our initial results suffered from this problem before we eliminated overlap in target words across folds.

For the original AVER, the correlation between predicted and actual mean human ratings is 0.180 for the linear model and 0.178 for the logistic model. The Area Under Curve (AUC) for the original AVER is 0.600, significantly better than the 0.5 expected from a random baseline.

The linear model predicts mean human ratings, so it optimizes the correlation of predicted to actual ratings. The logistic model classifies contexts as good or bad, so it optimizes the number of misclassified contexts. Consequently correlation is higher for the linear model, whereas AUC is higher for the logistic model.

Unfortunately, AVER.b fared considerably worse. Its predictions correlated with actual ratings at only .093, with AUC only 0.563. Accordingly we focus on the results for the original AVER.

Table 2 shows the original AVER linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at $p < .05$ (*), .005 (**), or .0005 (***).

Table 2: Coefficients of linear model for (original) AVER

Feature	Coefficient
WordBand	–.5691
Flesch-Kincaid Reading Ease	*** **.1220**
Flesch-Kincaid Grade	*** **.0627**
Average word length	*** **.0520**
Unigram logP(t)	* –1**.017**
Bigram logP(t w1 | __ w1)	*** **.0621**
Bigram logP(w1 t | w1 __)	** **.0188**
Trigram logP(t w1 w2 | __ w1 w2)	*** **–.0394**
Trigram logP(w1 t w2 | w1 __ w2)	.0070
Trigram logP(w1 w2 t | w1 w2 __)	*** **–.0053**
4gram logP(t w1 w2 w3 | __ w1 w2 w3)	.0088
4gram logP(w1 t w2 w3 | w1 __ w2 w3)	.0213
4gram logP(w1 w2 t w3 | w1 w2 __ w3)	–.0109
4gram logP(w1 w2 w3 t | w1 w2 w3 __)	*** **.0398**
5gram logP(t w1 w2 w3 w4 | __ w1 w2 w3 w4)	* **–.0297**
5gram logP(w1 t w2 w3 w4 | w1 __ w2 w3 w4)	–.0002
5gram logP(w1 w2 t w3 w4 | w1 w2 __ w3 w4)	.0193
5gram logP(w1w2w3 t w4 | w1 w2 w3 __ w4)	* **–.0283**
5gram logP(w1 w2 w3 w4 t | w1 w2 w3 w4 __)	.0017
Co-occurrence (DISCO S1)	*** **.0340**
Distributional Similarity (DISCO S2)	*** **.0674**
Intercept	*** **2.5079**

As Table 2 shows, unigram log probability of the target word was by far the strongest predictor of human ratings, and negative:

contexts for rarer words get lower ratings, which may reflect that the less frequently the target word appears in the Google *n*-grams corpus, the less likely it is to have good example contexts on the web. As expected, Reading Ease is a positive predictor: readable example contexts are likelier to help students. Surprisingly, the coefficients for word length and grade level are positive even though in isolation they correlate negatively with ratings. Perhaps they reflect positive effects exposed after other predictors account for the negative effects, or are simply artifacts of including correlated predictors in the model. Several *n*-gram based metrics of local predictability in the form of conditional probability of the target given the surrounding context are significant, but it is not clear why some are positive and others are negative. Fewer features based on longer *n*-grams are significant, presumably due to sparseness in the corpus. Finally, both topicality indicators are significant positive predictors: contexts relevant to a target word are likelier to be nutritious for learning it.

Although AVER.b's results were worse, they are easier to interpret, and differ from the original AVER. Table 3 shows AVER.b linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at $p < .05$ (*) or .0005 (***); one feature is suggestive at $p < .1$ (.).

Table 3: Coefficients of linear model for AVER.b

Feature	Coefficient	
WordBand	***	**0.0508**
Flesch-Kincaid Reading Ease	***	**0.0567**
Flesch-Kincaid Grade	*	**0.0328**
Average word length	*	-0.0199
Unigram logP(t)		0.0052
logP(t w1 w2 w3 w4 | __ w1 w2 w3 w4)	***	**0.0241**
logP(w1 t w2 w3 w4 | w1 __ w2 w3 w4)		-0.0039
logP(w1 w2 t w3 w4 | w1 w2 __ w3 w4)	***	**0.0415**
logP(w1w2w3 t w4 | w1 w2 w3 __ w4)	.	-0.0152
logP(w1 w2 w3 w4 t | w1 w2 w3 w4 __)	***	**0.0321**
Co-occurrence (DISCO S1)	***	**0.0483**
Distributional Similarity (DISCO S2)		0.0031
Intercept	***	**2.5823**

For AVER.b, WordBand is significant and Unigram is not, just the opposite of the original AVER. One reason may be that the AVER.b's context probabilities back off to unigram probability for the 8%-9% of 2-word contexts not listed in the bigram table. Reading Ease, Grade, and Word Length are significantly positive in both models. The five context probabilities show a striking pattern: the first, middle, and last positions in a 5-word context are highly predictive, whereas the other two are not. One candidate explanation is that target words tend to be adjacent to function words that provide much less specific information about them. However, the five features have similar correlations with Turker ratings, ranging from 0.038 to 0.065. A simpler explanation is that successive contexts make correlated predictions, and regression assigns the shared variance to just one.

Finally, DISCO S1 was highly significant in both models, but DISCO S2 was significant in the original AVER but not AVER.b. It is not obvious how to explain this difference based on the difference in representation of local context features, i.e., how backoff would steal variance from distributional similarity.

To compare the cross-validation results for the original AVER to a random baseline, Figure 1 shows the ROC for the percentage of good contexts (rated 3 or above) accepted against the percentage of bad (rated below 3) contexts accepted, as the acceptance threshold on the logistic model's output probability varies.

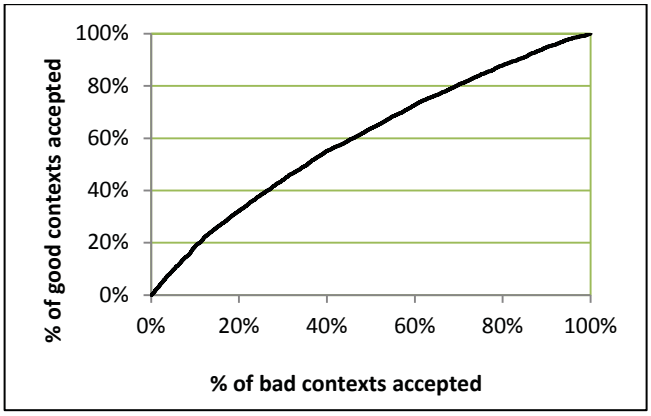


Figure 1: ROC curve for % good vs. % bad contexts accepted

Figure 2 plots the percentages of all the good and bad contexts accepted as the probability threshold decreases from 0.8.

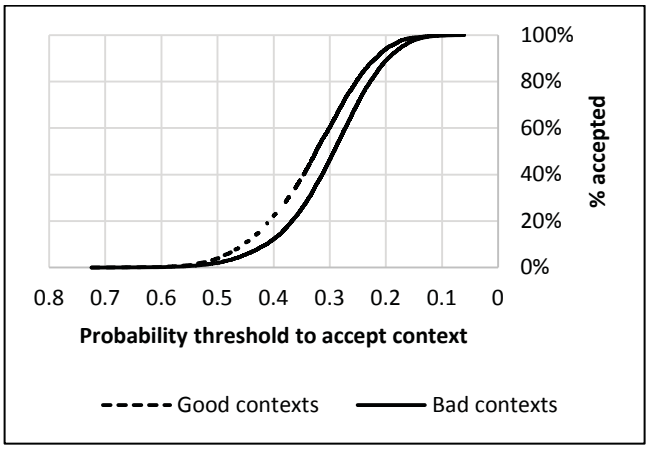


Figure 2: % of contexts accepted vs. probability threshold

As Figure 3 shows, the difference in percentages peaks at 15.2%:

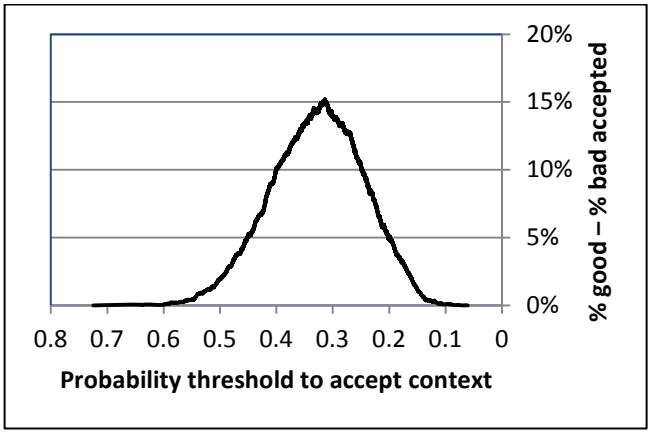


Figure 3: % good – % bad vs. probability threshold

However, bad contexts outnumber good ones, so even when the percentage accepted out of all the good contexts exceeds the percentage accepted out of all the bad contexts, the accepted contexts contains a higher percentage of bad than good contexts, and this imbalance worsens as the threshold decreases, as Figure 4 shows.

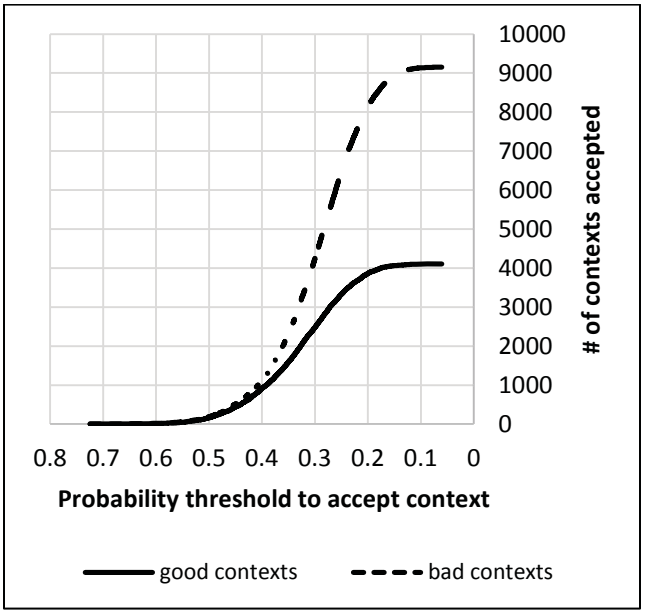


Figure 4: # of contexts accepted vs. probability threshold

As Figure 5 shows, at a threshold of 0.476, the ratio of good to bad contexts reaches a local peak of 0.911 – over twice as high as 0.449, the overall baseline ratio of good contexts to bad contexts. However, at such a high threshold, only 4.4% of the contexts are accepted: 278 (6.8%) of the 4107 good contexts and 305 (3.3%) of the 9150 bad contexts. Thus there is a tradeoff between the number and quality (% good) of the accepted contexts.

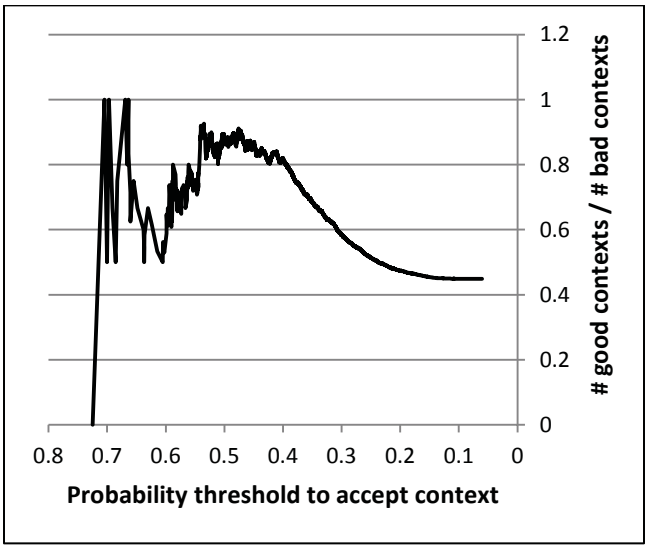


Figure 5: Ratio of good to bad contexts accepted

Visualizing the accuracy of the predicted ratings requires a different type of plot because predicting ratings is not a classification task**.** Accordingly, Figure 6 shows the distribution of errors in rating good and bad contexts as a histogram of predicted minus actual ratings, binned to the nearest 0.1. Figure 6 reflects the fact that there are many more bad than good contexts. It shows that almost all the errors in ratings are less than 1 in size.

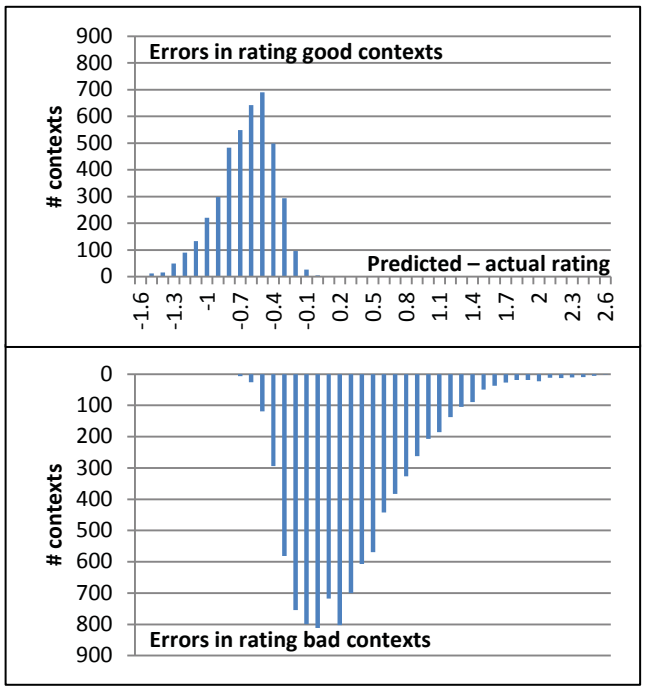


Figure 6: Histogram of errors in rating contexts

7. CONCLUSION

This paper presented and evaluated two models for predicting human ratings of example contexts for learning vocabulary. In contrast to prior work that used manually specified, explicitly operationalized criteria to evaluate contexts, both models approximate the implicit criteria underlying human judgments. Given the wide range of phenomena in language, the diversity of criteria that affect the nutritiousness of example contexts, and humans' limited ability to articulate these criteria explicitly and operationalize them precisely, models trained on human ratings have the potential to surpass hand-crafted models, just as machine learning has surpassed hand-crafted classifiers in other domains.

The AVER system reported here is just an initial step toward this goal: it rates contexts reliably more accurately than chance, but not by very much. Its features are shallow, based on local or bag-of-words statistics rather than deeper linguistic structures such as dependency graphs. Future work should develop more sophisticated features. Our analysis of example contexts with large discrepancies between actual and predicted ratings exposed some promising syntactic and semantic features, informed by human understanding of what makes particular contexts useful to learners or not.

Second, supervised learning from labeled data is only as good as the quality of the labels. The larger project of which this work is a part has already revised the training and selection of raters. However, even expert labels are only a proxy for what actually helps real students. Definitive labels should be grounded empirically in data on how much different students learn about different words from different example contexts. To be practical, this approach will require considerable amounts of data – even more so if it tries to model individual differences among students, not just what works well overall on average.

Third, we rated example contexts in isolation, but learning a word's meaning requires encountering it in diverse contexts, not just repeated encounters in the same context, because students learn different aspects from different contexts. Optimizing the entire sequence of encounters will require identifying what those different aspects are, what sorts of contexts help in learning which aspects, and how learning is affected by their order and how they are related.

Besides accelerating the practical task of selecting good example contexts to teach vocabulary, machine-learned models may eventually shed new light on what properties make example contexts nutritious for learning vocabulary, thereby improving our understanding of human vocabulary learning and instruction.

8. ACKNOWLEDGMENTS

The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A130467 ("Developing an Online Tutor to Accelerate High School Vocabulary Learning") to University of South Carolina (Suzanne Adlof, PI) and its subcontracts to Carnegie Mellon University (Jack Mostow, PI), and University of Pittsburgh (Charles Perfetti, PI). The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education. We thank DictionarySquared founder Adam Kapelner for the contexts, Margaret McKeown for expert ratings, Suzanne Adlof and Julie Byard for Turker ratings, and the reviewers for helpful comments.

9. REFERENCES

[1] Stanovich, K., R. West, and A.E. Cunningham. Beyond phonological processes: Print exposure and orthographic processing**.** In S. Brady and D. Shankweiler, Editors, ***Phonological Processes in Literacy***. Lawrence Erlbaum Associates: Hillsdale, NJ, 1992.

[2] Baumann, J.F., E.J. Kame'enui, and G.E. Ash. Research on vocabulary instruction: Voltaire redux**.** In J. Flood, et al., Editors, ***Handbook of research on teaching the English language arts***, 752-785. Erlbaum & Associates: Mahwah NJ, 2003.

[3] Graves, M.F. Vocabulary learning and instruction**.** In E.Z. Rothkopf, Editor, ***Review of Research in Education***, 91-128 1986.

[4] Mezynski, K. Issues concerning the acquisition of knowledge: Effects of vocabulary training on reading comprehension**.** ***Review of Educational Research***, 1983. ***53***: p. 253-279.

[5] Stahl, S.A. and M.M. Fairbanks. The effects of vocabulary instruction: A model-based meta-analysis**.** ***Review of Educational Research***, 1986. ***56***(1): p. 72-110.

[6] Graves, M.F. A Vocabulary Program to Complement and Bolster a Middle-Grade Comprehension Program**.** In B.M. Taylor, M.F. Graves, and P. van den Broek, Editors, ***Reading for Meaning: Fostering Comprehension in the Middle Grades. Language and Literacy Series***, 116-135. International Reading Association: Newark, DE, 2000.

[7] Biemiller, A. and C. Boote. An effective method for building meaning vocabulary in primary grades**.** ***Journal of Educational Psychology***, 2006. ***98***(1): p. 44-62.

[8] Stahl, S.A. and W.E. Nagy. ***Teaching Word Meanings***. Literacy Teaching Series. 2006, Mahwah, NJ: Lawrence Erlbaum Associates. ix+220.

[9] Beck, I.L., M.G. McKeown, and L. Kucan. ***Bringing Words to Life: Robust Vocabulary Instruction***. 2002, NY: Guilford.

[10] Pavlik Jr., P.I. and J.R. Anderson. Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect**.** ***Cognitive Science***, 2005. ***29***(4): p. 559-586.

[11] Aist, G.S. Helping Children Learn Vocabulary During Computer-Assisted Oral Reading**.** ***Educational Technology and Society***, 2002. ***5***(2): http://ifets.ieee.org/periodical/vol_2_2002/aist.html.

[12] Reinking, D. and S.S. Rickman. The effects of computer-mediated texts on the vocabulary learning and comprehension of intermediate-grade readers**.** ***Journal of Reading Behavior***, 1990. ***22***(4).

[13] Brown, J. and M. Eskenazi. Retrieval of Authentic Documents for Reader-Specific Lexical Practice. ***Proceedings of InSTIL/ICALL 2004 Symposium on Computer Assisted Learning***, paper 006. 2004. Venice, Italy.

[14] Liu, L., J. Mostow, and G.S. Aist. Generating Example Contexts to Help Children Learn Word Meaning**.** ***Journal of Natural Language Engineering***, 2013. ***19***(2): p. 187-212.

[15] Mostow, J. and W. Duan. Generating Example Contexts to Illustrate a Target Word Sense**.** In ***Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications***, J. Tetreault, J. Burstein, and C. Leacock, Editors. 2011, Association for Computational Linguistics, Stroudsburg, PA: Portland, OR, p. 105-110. At http://aclweb.org/anthology-new/W/W11/W11-14.pdf.

[16] Franz, A. and T. Brants. All Our *N*-gram are Belong to You**.** 2006. At http://googleresearch.blogspot.com/2006/08/all-our-*n*-gram-are-belong-to-you.html.

[17] Kolb, P. Disco: A multilingual database of distributionally similar words. ***Proceedings of KONVENS-2008 (Konferenz zur Verarbeitung natürlicher Sprache)*** 2008. Berlin.

[18] BNC Consortium. The British National Corpus, version 3 (BNC XML Edition)**.** 2007, Oxford University Computing Services. At http://www.natcorp.ox.ac.uk/.

[19] Surdeanu, M., J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. ***Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations***, 55-60. 2014. Baltimore, MD.

[20] Goldberg, Y. and J. Orwant. A dataset of syntactic-ngrams over time from a very large corpus of english books. ***Second Joint Conference on Lexical and Computational Semantics (* SEM)***, 241-247. 2013.

[21] Jang, H. and J. Mostow. Inferring Selectional Preferences from Part-of-Speech *N*-grams**.** In ***Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics***. 2012: Avignon, France, p. 377-386.

Mining a Written Values Affirmation Intervention to Identify the Unique Linguistic Features of Stigmatized Groups

Travis Riddle [*], Sowmya Sree Bhagavatula[1], Weiwei Guo[1], Smaranda Muresan[1], Geoff Cohen[2], Jonathan E. Cook[3], and Valerie Purdie-Vaughns[1]

[1]Columbia University
[2]Stanford University
[3]Pennsylvania State University

ABSTRACT

Social identity threat refers to the process through which an individual underperforms in some domain due to their concern with confirming a negative stereotype held about their group. Psychological research has identified this as one contributor to the underperformance and underrepresentation of women, Blacks, and Latinos in STEM fields. Over the last decade, a brief writing intervention known as a values affirmation, has been demonstrated to reduce these performance deficits. Presenting a novel dataset of affirmation essays, we address two questions. First, what linguistic features discriminate gender and race? Second, can topic models highlight distinguishing patterns of interest between these groups? Our data suggest that participants who have different identities tend to write about some values (e.g., social groups) in fundamentally different ways. These results hold promise for future investigations addressing the linguistic mechanism responsible for the effectiveness of values affirmation interventions.

Keywords

Interventions, Natural Language Processing, Achievement Gap

1. INTRODUCTION

In the American education system, achievement gaps between Black and White students and between male and female students persist despite recent narrowing. This is true in STEM fields in particular, with the underachievement leading in turn to problems with underemployment and underrepresentation more generally. Women, for example, make up a scant 28% of the STEM workforce [1].

While we acknowledge that the reasons for underachievement and underrepresentation are numerous and complex, *social identity threat* has consistently been shown to be one factor which contributes to these problems and features a psychological basis [32]. Social identity threat refers to the phenomenon in which an individual experiences stress due to concerns about confirming a negative stereotype held about his or her social group. For instance, Black students are stereotyped to be less capable in academic settings than White students. Therefore, a Black student who is aware of this stereotype may feel psychologically threatened, leading to changes in affect, physiology, and behavior[17, 35, 27, 5].

The description of a psychological process that partly accounts for these achievement gaps opens the door to possible psychological interventions. Indeed, a brief, relatively simple intervention derived from self-affirmation theory known as a *values affirmation* has been shown to diminish these achievement gaps - especially when delivered at key transitional moments, such as the beginning of an academic year [6, 4]. The values-affirmation intervention instructs students to choose from a series of values, and then reflect on why this value might be important to them. The intervention draws on self-affirmation theory, which predicts that a fundamental motivation for people is to maintain self-integrity, defined as being a good and capable individual who behaves in accordance with a set of moral values [31].

Accumulating evidence indicates that this intervention is effective in reducing the achievement gap. For instance, students who complete the intervention have shown a blunted stress response [8] and improved academic outcomes longitudinally [4], as well as in the lab [13, 26]. There is also evidence that these affirmations reduce disruptive or aggressive behavior in the classroom [33, 34].

In short, research has definitively shown that values affirmations can reduce achievement gaps. However, the content of the essays themselves has not been as thoroughly examined. While some studies have examined the content of expressive writing for instances of spontaneous affirmations [7], or examined affirmations for instances of certain pre-defined themes (e.g., social belonging [28]), these efforts have been on a relatively small scale, and have been limited by the usual constraints associated with hand-annotating (e.g., experimenter expectations, annotator bias, or excessive time

[*]tar2119@columbia.edu; Corresponding Author

requirements).

The goal of this paper is to explore the *content of values affirmation essays* using *data mining techniques*. We explore the differences in the content of affirmation essays as a function of ethnic group membership and gender. We are motivated to address these questions because ethnicity and gender, in the context of academic underperformance and the affirmation intervention, are categorical distinctions of particular interest. Identifying as Black or as a woman means that one is likely to contend with negative stereotypes about intelligence, which in turn puts the individual at risk of experiencing the negative effects of social identity threat. The content of the essays produced by individuals under these different circumstances could lead to insights on the structure of threat or the psychological process of affirmation. Additionally, we hope to eventually use information from this initial study to create affirmation prompts which are tailored to individual differences. That is, it may be beneficial to structure the values-affirmation in different ways depending on the particular threatening context or identity of the writer.

We will explore these issues from two different perspectives. First, we investigate the latent topics of essays using Latent Dirichlet Allocation (LDA) [2], which is a generative model that uncovers the thematic structure of a document collection. Using the distribution of topics in each essay, we will present examples of topics which feature strong and theoretically interesting between-group differences. Second, we approach the question of between-group differences in text as a classification problem. For instance, given certain content-based features of the essays (e.g., topics, n-grams, lexicon-based words), how well can we predict whether an essay was produced by a Black or White student? This approach also allows us to examine those features which are the most strongly discriminative between groups of writers. Finally, classification will allow us to closely compare the relative strength of each model's features with respect to differences between groups.

2. DATA

Our data come from a series of studies conducted on the effectiveness of values affirmations. For the datasets that have resulted in publications, detailed descriptions of the subjects and procedures can be found in those publications [4, 5, 27, 28]. The unpublished data follow nearly identical procedures with respect to the essay generation.

As an illustrative example of the essay generation process, we describe the methods from Cohen et. al [4]. This study, conducted with seventh-graders, featured a roughly equal number of Black and White students who were randomly assigned to either the affirmation condition or a control condition. The affirmation intervention was administered in the student's classrooms, by teachers who were blind to condition and hypothesis. Near the beginning of the fall semester, students received closed envelopes from their teachers, who presented the work as a regular classroom exercise. Written instructions inside the envelope guided students in the affirmation condition to chooose their most important values (or, in study 2, their top two or three most important values) from a list (athletic ability, being good at art, being smart or getting good grades, creativity, independence, living in the moment, membership in a social group, music, politics, relationships with friends or family, religious values, and sense of humor), while control students were instructed to select their least important value (two or three least important values in study 2). Students in the affirmation condition then wrote about why their selected value(s) are important to them, while students in the control condition wrote about why their selected values might be important to someone else. All students quietly completed the material on their own.

The other samples in our data include both lab and field studies and feature methods largely similar to those just described. Across all studies, participants completing the affirmation essays are compared with students who do not suffer from social identity threat as well as students who complete a control version of the affirmation. Our datasets feature students of college age, as well as middle school students. Below we show two examples of *affirmation essays* (one from a college student and one from a middle school student) and a *control essay* (middle school student):

> **Affirmation Essay (college student):** My racial/ethnic group is most important to me when I am placed in situations that are alienating or dangerous or disrespectful. Since coming to Yale a school much larger than my former school where I feel my minority status that much more sharply or feel like people are judging me because I have dark skin I have placed a much higher value on being black. I work for the Af-Am House. I am involved in Black groups and most of my friends are Black. But often being black holds me down and depresses me because people are surprised at how much like them I can be and I dont think Im pretty. Its stressful to have to avoid stereotypes like being late or liking to dance or being sexual. I dont want people to put me in a box labeled black Girl 18. I am my own person.

> **Affirmation Essay (middle school student:)** Being smart and getting good grades is important to me because it is my path to having a succesful life. Independence is also important because I don't want to be like everybody else. I want to be special in my own way. I want to be different.

> **Control Essay:** I think that being good in art can be important to someone else who likes and enjoys art more than I do. I also think this because there are people who can relate and talk about art by drawing and stuff like that but I don't.

In total, we were able to obtain 6,704 essays. Of these, our analyses included all essays which met the following criteria:

1. The essay was an *affirmation* essay (not control). We opted to exclude control essays because the psycholog-

ical process behind the generation of a control essay is fundamentally different from the process that generates an affirmation essay. We are interested in the *affirmation* process, and including control essays in a topic model, for instance, would only add noise to the signal we are interested in exploring.

2. The writing prompt did not deviate (or deviated only slightly) from the writing prompt most widely used across various studies [4]. For example, most of the essays used prompts mentioned above (e.g., athletic ability, religious values, independence). We excluded prompts such as reflection on President Obama's election, since they are of a different nature.

Including only the essays which met the above criteria resulted in a final dataset of 3,097 essays. Given that some individuals wrote up to 7 essays over the period of their participation, the 3,097 essays came from 1,255 writers (425 Black, 473 White, 41 Asian, 174 Latino, 9 other, 83 unrecorded; 657 females, 556 males, 42 unrecorded). The majority of these writers (n = 655) were from a field study in which 8 cohorts of middle school students were followed over the course of their middle school years. The remainder were from several lab-based studies conducted with samples of college students. Before modeling, all essays were preprocessed by removing stop words and words with frequency counts under four. We also tokenized, lemmatized, and automatically corrected spelling using the jazzy spellchecker [11].

The essays varied in length (median number of words = 39, mean = 44.83, SD = 35.85). Some essays are very short (e.g., 2 sentences). As we describe in the next section, this posed some interesting opportunities to test different methods of modeling these essays, especially with regard to using topic models.

3. MODELS FOR CONTENT ANALYSIS

To explore the differences in the content of affirmation essays as a function of ethnic group membership and gender we used several methods to model essay content.

Latent Dirichlet Allocation (LDA). Graphical topic models such as LDA [2] have seen wide application in computational linguistics for modeling document content. Such topic models assume that words are distributed according to a mixture of topics and that a document is generated by selecting a topic with some mixture weight, generating a word from the topic's word distribution, and then repeating the process. LDA specifies a probabilistic procedure by which *essays* can be generated: the writer chooses a topic z_n at random according to a multinomial distribution (θ), and draws a word w_n from $p(w_n|z_n, \beta)$, which is a multinomial probability conditioned on the topic z_n ($\theta \sim Dir(\alpha)$). The topic distribution θ describes the portion of each topic in a document. One drawback of the current LDA framework is that it assumes equal contribution of each word to the topic distribution of a document θ. Since many of our writers tended toward using repetitive language (e.g., miming the essay prompt), we used a modified version of LDA to model our essays, which uses a tf-idf matrix instead of the standard word-count matrix [21]. This allows words that are more unique in their usage to take on greater weight in the topic model. We settled on a model with 50 topics, as this provided a good fit to our data, and topics with good subjective interpretability. Given that a primary goal of our analysis was to investigate the topics, we prioritized interpretable topics over statistical fit when necessary. Figure 1 shows the affirmation essay written by the college student given in Section 2, where words are highlighted to show their topic assignments. This example includes three topics, one of which is clearly related to ethnic group (red text), while the other two are somewhat more ambiguous. Section 4 shows some of the learned topics, an analysis of the topic distributions as a function of gender and race, and the results of using the topic distributions as additional features for classification experiments (gender, ethnicity, and gender-ethnicity).

My racial/ethnic group is most important to me when I am placed in situations that are alienating or dangerous or disrespectful. Since coming to Yale a school much larger than my former school where I feel my minority status that much more sharply or feel like people are judging me because I have dark skin I have placed a much higher value on being black. I work for the Af-Am House. I am involved in Black groups and most of my friends are Black. But often being black holds me down and depresses me because people are surprised at how much like them I can be and I dont think Im pretty. Its stressful to have to avoid stereotypes like being late or liking to dance or being sexual. I dont want people to put me in a box labeled black Girl 18. I am my own person.

Figure 1: An example essay from a college-aged writer. Words have been highlighted to show their topic assignments

Weighted Textual Matrix Factorization (WTMF). Topic models such as LDA [2] have been successfully applied to relatively lengthy documents such as articles, web documents, and books. However, when modeling short documents (e.g., tweets) other models such as Weighted Textual Matrix Factorization (WTMF) [10] are often more appropriate. Since most of our essays are relatively short (2-3 sentences), we use WTMF as an additional method to model essay content. The intuition behind WTMF is that it is very hard to learn the topic distribution only based on the limited observed words in a short text. Hence Guo and Diab [10] include unobserved words that provide thousands more features for a short text. This produces more robust low dimensional latent vector for documents. However, while WTMF is developed to model latent dimensions (i.e., topics) in a text, a method for investigating the most frequent words of these latent dimensions is not apparent (unlike LDA). We therefore use this content analysis method only for the classification tasks (gender, ethnicity, gender-ethnicity), with the induced 50 dimensional latent vector as 50 additional features in classification (Section 4).

Linguistic Inquiry and Word Count (LIWC). Pennebaker et al.'s LIWC (2007) dictionary has been widely used both in psychology and computational linguistics as a method for content analysis. The LIWC lexicon consists of a set of 64

Table 1: Top 10 words from select LDA topics

Topic3	Topic22	Topic33	Topic43	Topic47
relationship	time	group	religion	religious
life	spring	black	church	god
feel	play	white	religious	faith
independent	hang	racial	god	religion
family	talk	identify	treat	jesus
support	help	race	sunday	believe
time	friend	ethnic	believe	belief
friend	family	certain	famous	church
through	homework	culture	stick	christian
help	school	history	lord	earth

word categories grouped into four general classes organized hierarchically: 1) Linguistic Processes (LP) [e.g., Adverbs, Pronouns, Past Tense, Negation]; 2) Psychological Processes (PP) [e.g., Affective Processes [Positive Emotions, Negative Emotions [Anxiety, Anger, Sadness]], Perceptual Processes [See, Hear, Feel], Social Processes, etc]; 3) Personal Concerns (PC) [e.g., Work, Achievement, Leisure]; and 4) Spoken Categories (SC) [Assent, Nonfluencies, Fillers]. LIWC's dictionary contains around 4,500 words and word stems. In our analysis we used LIWC's 64 categories as lexicon-based features in the classification experiments (Section 4).

4. RESULTS

One of our primary questions of interest is whether we can discover between-group differences in the content of the essays. In order to examine this idea in a straightforward way, we limit the analyses to only those individuals who identified as Black or White (2,392 essays from 897 writers). While there are stereotypes suggesting that Asians and Latinos should perform well and poorly in academic domains, respectively, many individuals in our samples who identify with these groups are born in other countries, where the nature of prevailing stereotypes may be different. This is not true to the same extent of individuals who identify as Black or White. We thus exclude Asians and Latinos (as well as those who identified as "other" or declined to answer) for our between-group differences analyses and classification experiments. Inferential analyses were conducted using R [20], and figures were generated using the ggplot2 package [36].

4.1 Interpreting Topic Models

We first describe the results of using LDA to see whether we can detect topics that feature strong and theoretically interesting between-group differences. Accurately interpreting the meaning of learned topics is not an easy process [14] and more formal methods are needed to qualitatively evaluate these topics. However, our initial investigation suggests that participants use common writing prompts to write about values in different ways, depending on the group to which they belong.

Table 1 provides the top 10 words from several learned LDA topics[1]. Manually inspecting the topics, we noticed that LDA not only learned topics related to the values given, but it seemed to be able to learn various aspects related to these

[1]As noted in section 3, we are unable to investigate WTMF models in the same fashion.

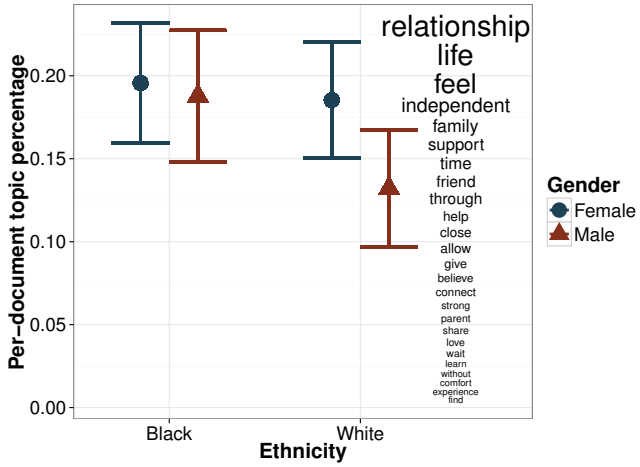


Figure 2: Topic3: Most prominent topic. Points represent fixed effect estimates. Error bars represent represent +/- 1.96 standard errors. Word size represents weighting in the topic

values. For example, Topic43 and Topic47 both relate to religious values but Topic43 refers to religion as it pertains to elements of the institution (including words such as church, sunday, and catholic), while Topic47 seems to focus more on the content of faith itself (indicated by words such as faith, jesus, and belief). A similar interpretation can be given to Topic3 and Topic22 — they both refer to relationship with family and friends, but one focuses on the support and help aspect (Topic3), while the other seems to refer to time spent together and hanging out (Topic22). Finally, Topic33 shows an example where the topic learned is about ethnic group, even if ethnicity was not a specific value given as a prompt (rather the more general value of 'membership in a social group' was given). Figure 1 shows an example of an essay and the word-topic assignments, where Topic33 is one of the topics (ethnic group, shown in red).

In order to identify interesting between-group differences in topic distributions, we fit a series of mixed-effects linear regressions, with each of the 50 topics as the outcomes of interest. For each model, we estimated effects for gender, ethnicity, and the interaction between the two. For the random effects component, we allowed the intercept to vary by writer. Across the 50 models and excluding the intercept, we estimated a total of 150 effects of interest. Of these, 23 reached the threshold for statistical significance. This proportion is greater than would be expected by chance ($p < .01$). Having established that there are real and meaningful between-groups differences, we more closely examined topics which had theoretically interesting insights.

For example, Figure 2 shows the most frequent words from the most prominent topic (Topic3; relationships with family and friends as basis of support/help) across all essays, along with differences between groups. The model for this topic yielded marginal effects of gender ($B = .02$, $SE = .01$, $p = .08$), with female writers devoting a greater proportion of their writing to the topic ($M = .12$, $SD = .27$) than males ($M = .09$, $SD = .24$). There was also a marginal effect of

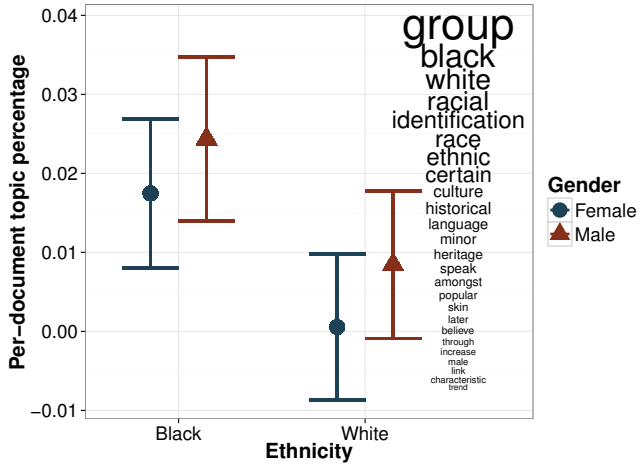


Figure 3: Topic33: effect of ethnicity. Points represent fixed effect estimates. Error bars represent represent +/- 1.96 standard errors. Word size represents weighting in the topic

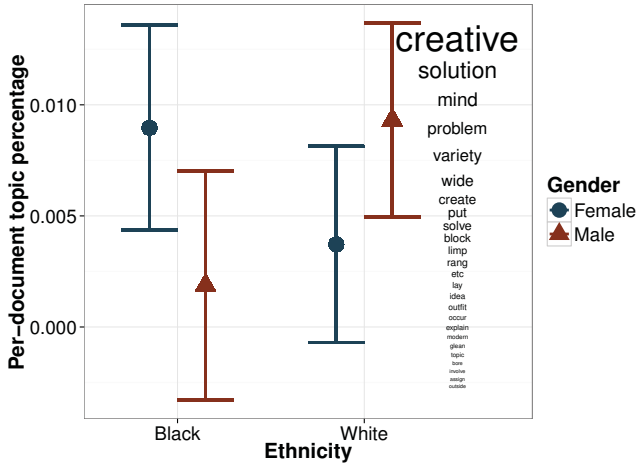


Figure 4: Topic23: Interaction between Gender and Ethnicity. Points represent fixed effect estimates. Error bars represent represent +/- 1.96 standard errors. Word size represents weighting in the topic

ethnicity, ($B = .02$, $SE = .01$, $p = .10$), with black writers ($M = .11$, $SD = .26$) devoting more of their writing to the topic than white ($M = .10$, $SD = .25$) writers.

There were also topics which strongly discriminated between ethnicities. Figure 3 presents findings from one such topic (Topic33; ethnic group). The model for this topic revealed the expected main effect of ethnicity ($B = .008$, $SE = .02$, $p < .01$), with black writers devoting a greater proportion of their writing to the topic ($M = .01$, $SD = .07$) than white writers ($M = .003$, $SD = .03$).

The LDA model also estimated topics that were utilized differently by black and white writers, depending on if they happened to be males or females. For instance, Figure 4 presents a topic which is related to problem-solving. Modeling this topic showed that the interaction between gender and ethnicity was significant ($B = .003$, $SE = .01$, $p < .01$). Specifically, for black writers, women wrote more about this topic ($M = .009$, $SD = .07$) than males did ($M = .001$, $SD = .02$, $p < .05$). For white writers, the difference is in the opposite direction, and marginally significant, with males using more of their writing on this topic ($M = .009$, $SD = .08$) than women ($M = .004$, $SD = .03$, $p = .08$). Similarly, the difference for black and white males is statistically significant ($p < .05$), whereas the difference is reversed and marginal for black and white females ($p = .11$).

The findings from the LDA topic modeling show that there are between-group differences emerging from the affirmation essays. To investigate further, in the next section we present the results of a study where we approach the question of between-group differences as a classification problem.

4.2 Classification:Gender, Ethnicity, Gender-Ethnicity

Given certain content-based features of the essays (e.g., distribution of topics, LIWC categories, n-grams), these experiments aim to classify essays based on the writer's ethnicity and/or gender: Black vs. White (Ethnicity classification), Female vs. Male (Gender classification), and Black-Male vs White-Male and Black-Female vs. White-Female (Ethnicity-Gender classification). In all classification experiments we use a linear Support Vector Machine (SVM) classifier implemented in Weka (LibLINEAR) [9]. We ran 10-fold cross validation and for all results we report weighted F-1 score. As features we used TF-IDF (words weighted by their TF-IDF values)[2]; LDA (topic distributions are used as additional features); WTMF (the 50 dimensional latent vector used as 50 additional features) and LIWC (LIWC's 64 word categories are used as features).

The classification results are displayed in Table 2. We notice that all features give similar performance per classification task. In general, the results were better for the gender classification task (best results 74.09 F1 measure), while the worse results seems to be for the ethnicity classification (best result 66.37 F1). None of the classification tasks showed significant differences as a function of the included features ($p > .05$).

However, the aspect we were more interested in was to analyze the most discriminative features for each classification task with the hope of discovering interesting patterns for between-groups differences. The top 10 discriminating features from each classification type on the TF + LDA + LIWC features are presented in Table 3. There are several interesting observations when analyzing these results. First, supporting the results of the classification experiment, we see that unigrams feature prominently. We also note that LIWC features are largely missing from the top ten, with the only exception being the 10th feature for males in the gender classification. LDA topics, on the other hand, appear as strongly distinguishing in 3 of the 4 classification tasks. Further, in terms of content, the discriminative features sup-

[2]We experimented with presence of n-grams but using TF-IDF gives better results.

Table 2: SVM Results - cell contents are number of P/R/F1

Features	Classification			
	Gender	Ethnicity	Bl vs Wh Female	Bl vs Wh Male
TF-IDF	73.38/73.38/73.33	71.34/67.91/65.13	73.43/69.70/67.97	75.26/70.76/67.29
TF-IDF + LDA	73.48/73.46/73.40	**70.54/68.41/66.37**	73.29/69.62/67.90	74.72/70.85/67.63
TF-IDF + WTMF	73.52/73.46/73.37	71.72/68.00/65.11	**73.11/70.02/68.55**	74.62/70.59/67.23
TF-IDF+LIWC	74.07/74.0/73.92	72.07/68.08/65.10	73.49/69.78/68.07	75.20/70.85/67.45
TF-IDF+LDA+LIWC	**74.09/74.09/74.04**	71.38/68.58/66.24	73.49/69.78/68.07	**74.98/71.02/67.82**

Table 3: Most discriminative features from classifiers with TF-IDF+LDA+LIWC as features

Gender		**Ethnicity**	
Female	Male	Black	White
softball	verry	race	Topic15-relationship, creative
jump	available	result	Topic25-music, play, enjoy
swim	football	heaven	younger
happier	Topic26-play, soccer	barely	less
horse	score	disappoint	weird
cheerleader	language	romantic	Topic17-humor, sense, laugh
doctor	lazy	NBA	larger
Topic14-music, relax	moreover	outdoor	rock
boyfriend	baseball	africa	tease
reason	LIWC27-affect	double (game double dutch)	heavy
Females		**Males**	
Black	White	Black	White
double (game double dutch)	decorate	Topic22-spring, hangout	Topic25-music, play, enjoy
above	rock	NBA	Topic17-humor, sense, laugh
ill	guitar	race	Topic2-reply, already, told
race	peer	head	larger
thick	horse	motive	sit
south	handle	health	cheer
option	grandparents	apart	rock
lord	saxaphone	phone	skate
result	crowd	award	handy
york	less	famous	holiday

port some of the results from the topic model analysis. For instance, topic 33 (ethnic group) is the most discriminative, non-unigram feature for ethnicity, and is the 56th most strongly associated feature with Black writers overall. It is also the most discriminative, non-unigram feature for the female-ethnicity classification, as the 44th most strongly associated feature with Black female writers. However, this topic does not show up for the Black vs White male classification. The topic results (Figure 3) also indicate a somewhat stronger relationship for Black vs. White Females.

We also notice that there are strong effects related to sports. In particular, some of the most discriminative features are consistent with social expectations regarding participation in various types of sports. Females, for instance, are more likely to write about softball, swimming, and jumping rope, whereas males are more likely to write about football and baseball. Similar differences can be seen for ethnicity (NBA, double dutch), and gender-ethnicity classifications (females: double dutch, horse; males: NBA, skate).

5. RELATED WORK

As mentioned in the introduction, there have been some smaller-scale investigations into the content of affirmation essays. For instance, Shnabel et al.[28] hand-annotated a subset of the data presented here for presence of social belonging themes. They defined social belonging as writing about an activity done with others, feeling like part of a group because of a shared value or activity, or any other reference to social affiliation or acceptance. Their results indicate that the affirmation essays were more likely to contain such themes than control essays, and that Black students who wrote about belonging themes in their affirmation essays had improved GPAs relative to those who did not write about social belonging. A subsequent lab experiment confirmed this basic effect and strengthened the hypothesized causal claim. The data here are consistent with the idea that social themes are a dominant topic in these essays. Indeed, the most prominent topic (Topic3) seems to be a topic that directly corresponds to social support (see Table 1). Further, even a cursory glance at the topics we have included here will show that references to other people feature prominently - a pattern that is also true for the topics we have not discussed in this paper.

One other finding of interest concerns the discriminative ability of LIWC. Only for the gender classification did LIWC categories appear among the discriminative features. There

are many studies that show gender differences in LIWC categories [25, 19, 24, 16], to say nothing of the broader literature on differences in language use between men and women [15, 12]. However, there is far less consistent evidence for differences in LIWC categories as a function of ethnicity [18]. That our results indicate features from LDA are more discriminative for ethnicity suggests the utility of a bottom-up approach for distinguishing between these groups. However, it should be noted that, in general, classification performance on ethnicity was not as good as classification on gender.

Finally, we also note that this is one of a small, but growing number of studies directly contrasting LIWC and LDA as text modeling tools [30, 22, 25]. While this other work tends to find that LDA provides additional information which results in improvements to classification performance in comparison to LIWC, our do not display this pattern. It is not clear why this may be, although we suspect that frequent misspellings present in our data could lead to some of the discrepancy.

6. CONCLUSIONS

We used data mining techniques to explore the content of a written intervention known as a *values affirmation*. In particular, we applied LDA to examine latent topics that appeared in students' essays, and how these topics differed as a function of whether the group to which the student belonged (i.e., gender, ethnicity) was subject to social identity threat. We also investigated between-groups differences in a series of classification studies. Our results indicate that there are indeed differences in what different groups choose to write about. This is apparent from the differences in topic distributions, as well as the classifier experiments where we analyzed discriminative features for gender, ethnicity and gender-ethnicity.

Why might individuals coping with social identity threat write about different topics than those who are not? Some literature shows that racial and gender identity can be seen as a positive for groups contending with stigma [29]. The model of optimal distinctiveness actually suggests that a certain degree of uniqueness leads to positive outcomes [3]. This suggests that if an individual from a stigmatized group perceives their identity to be unique, it may be a source of pride. In the current context, this could be reflected in an increase of writing devoted to the unique social group students are a part of (i.e., African American). On the other hand, there is some evidence that individuals downplay or conceal identities they perceive to be devalued by others [23]. This work would suggest that students in our data would choose to write about what they have in common with others. Our work here seems to provide some support for the former, but we have not addressed these questions directly, and so cannot make any strong claims.

Looking forward, we intend to investigate the relationship between essay content and academic outcomes. Do stigmatized students who write about their stigmatized group experience more benefit from the affirmation, as would be suggested by the optimal distinctiveness model? This work could provide data that speak to this issue. Furthermore, we hope to model the trajectory of how the writing of an individual changes over time, especially as a function of whether they completed the affirmation or control essays. Given that values affirmations have been shown to have long-term effects, and our data include some individuals who completed multiple essays, exploration of longitudinal questions about the affirmation are especially intriguing. We also intend to model the essays using supervised-LDA, which would allow us to jointly model the topics with the grouping information. Last but not least we plan to investigate whether there are differences between the middle school students and the college-level students.

7. ACKNOWLEDGMENTS

We would like to thank Robert Backer and David Watkins for assistance with this project. This work was supported in part by the NSF under grant DRL-1420446 and by Columbia University's Data Science Institute through a Research Opportunities and Approaches to Data Science (ROADS) grant.

8. REFERENCES

[1] Women, minorities, and persons with disabilities in science and engineering. Technical Report NSF 13-304, National Science Foundation, National Center for Science and Engineering Statistics, Arlington, VA., 2013.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] M. B. Brewer. The social self: On being the same and different at the same time. *Personality and Social Psychology Bulletin*, 17(5):475–482, 1991.

[4] G. L. Cohen, J. Garcia, N. Apfel, and A. Master. Reducing the racial achievement gap: A social-psychological intervention. *Science*, 313(5791):1307–1310, 2006.

[5] G. L. Cohen, J. Garcia, V. Purdie-Vaughns, N. Apfel, and P. Brzustoski. Recursive processes in self-affirmation: Intervening to close the minority achievement gap. *Science*, 324(5925):400–403, 2009.

[6] J. E. Cook, V. Purdie-Vaughns, J. Garcia, and G. L. Cohen. Chronic threat and contingent belonging: Protective benefits of values affirmation on identity development. *Journal of Personality and Social Psychology*, 102(3):479, 2012.

[7] J. D. Creswell, S. Lam, A. L. Stanton, S. E. Taylor, J. E. Bower, and D. K. Sherman. Does self-affirmation, cognitive processing, or discovery of meaning explain cancer-related health benefits of expressive writing? *Personality and Social Psychology Bulletin*, 33(2):238–250, 2007.

[8] J. D. Creswell, W. T. Welch, S. E. Taylor, D. K. Sherman, T. L. Gruenewald, and T. Mann. Affirmation of personal values buffers neuroendocrine and psychological stress responses. *Psychological Science*, 16(11):846–851, 2005.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear - a library for large linear classification, 2008. The Weka classifier works with version 1.33 of LIBLINEAR.

[10] W. Guo and M. Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational*

Linguistics: Long Papers-Volume 1, pages 864–872. Association for Computational Linguistics, 2012.

[11] M. Idzelis. Jazzy: The java open source spell checker, 2005.

[12] R. T. Lakoff. *Language and woman's place: Text and commentaries*, volume 3. Oxford University Press, 2004.

[13] A. Martens, M. Johns, J. Greenberg, and J. Schimel. Combating stereotype threat: The effect of self-affirmation on women's intellectual performance. *Journal of Experimental Social Psychology*, 42(2):236–243, 2006.

[14] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 490–499. ACM, 2007.

[15] A. Mulac, J. J. Bradac, and P. Gibbons. Empirical support for the gender-as-culture hypothesis. *Human Communication Research*, 27(1):121–152, 2001.

[16] M. L. Newman, C. J. Groom, L. D. Handelman, and J. W. Pennebaker. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3):211–236, 2008.

[17] H.-H. D. Nguyen and A. M. Ryan. Does stereotype threat affect test performance of minorities and women? A meta-analysis of experimental evidence. *Journal of Applied Psychology*, 93(6):1314, 2008.

[18] M. Pasupathi, R. M. Henry, and L. L. Carstensen. Age and ethnicity differences in storytelling to young children: Emotionality, relationality and socialization. *Psychology and Aging*, 17(4):610, 2002.

[19] J. W. Pennebaker and L. A. King. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6):1296, 1999.

[20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[21] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. *ICWSM*, 5(4):130–137, 2010.

[22] P. Resnik, A. Garron, and R. Resnik. Using topic modeling to improve prediction of neuroticism and depression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1348–1353. Association for Computational Linguistics, 2013.

[23] L. M. Roberts. Changing faces: Professional image construction in diverse organizational settings. *Academy of Management Review*, 30(4):685–711, 2005.

[24] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

[25] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

[26] J. R. Shapiro, A. M. Williams, and M. Hambarchyan. Are all interventions created equal? A multi-threat approach to tailoring stereotype threat interventions. *Journal of Personality and Social Psychology*, 104(2):277, 2013.

[27] D. K. Sherman, K. A. Hartson, K. R. Binning, V. Purdie-Vaughns, J. Garcia, S. Taborsky-Barba, S. Tomassetti, A. D. Nussbaum, and G. L. Cohen. Deflecting the trajectory and changing the narrative: How self-affirmation affects academic performance and motivation under identity threat. *Journal of Personality and Social Psychology*, 104(4):591, 2013.

[28] N. Shnabel, V. Purdie-Vaughns, J. E. Cook, J. Garcia, and G. L. Cohen. Demystifying values-affirmation interventions writing about social belonging is a key to buffering against identity threat. *Personality and Social Psychology Bulletin*, 39(5):663–676, 2013.

[29] T. B. Smith and L. Silva. Ethnic identity and personal well-being of people of color: a meta-analysis. *Journal of Counseling Psychology*, 58(1):42, 2011.

[30] A. Stark, I. Shafran, and J. Kaye. Hello, who is calling?: Can words reveal the social nature of conversations? In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 112–119. Association for Computational Linguistics, 2012.

[31] C. M. Steele. The psychology of self-affirmation: Sustaining the integrity of the self. *Advances in Experimental Social Psychology*, 21:261–302, 1988.

[32] C. M. Steele, S. J. Spencer, and J. Aronson. Contending with group image: The psychology of stereotype and social identity threat. *Advances in Experimental Social Psychology*, 34:379–440, 2002.

[33] S. Thomaes, B. J. Bushman, B. O. de Castro, G. L. Cohen, and J. J. Denissen. Reducing narcissistic aggression by buttressing self-esteem: An experimental field study. *Psychological Science*, 20(12):1536–1542, 2009.

[34] S. Thomaes, B. J. Bushman, B. O. de Castro, and A. Reijntjes. Arousing âĂIJgentle passionsâĂİ in young adolescents: Sustained experimental effects of value affirmations on prosocial feelings and behaviors. *Developmental Psychology*, 48(1):103, 2012.

[35] G. M. Walton and G. L. Cohen. Stereotype lift. *Journal of Experimental Social Psychology*, 39(5):456–467, 2003.

[36] H. Wickham. *ggplot2: Elegant graphics for data analysis*. Springer New York, 2009.

Automatic Classification of Question & Answer Discourse Segments from Teacher's Speech in Classrooms

Nathaniel Blanchard
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
nblancha@nd.edu

Sidney D'Mello
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
sdmello@nd.edu

Andrew M. Olney
University of Memphis
365 Innovation Drive
Memphis, TN 38152, USA
aolney@memphis.edu

Martin Nystrand
University of Wisconsin-Madison
685 Education Sciences
Madison WI, 53706-1475
mnystrand@ssc.wisc.edu

ABSTRACT

Question-answer (Q&A) is fundamental for dialogic instruction, an important pedagogical technique based on the free exchange of ideas and open-ended discussion. Automatically detecting Q&A is key to providing teachers with feedback on appropriate use of dialogic instructional strategies. In line with this, this paper studies the possibility of automatically detecting segments of Q&A in live classrooms based solely on audio recordings of teacher speech. The proposed approach has two steps. First, teacher utterances were automatically detected from the audio stream via an amplitude envelope thresholding-based approach. Second, supervised classifiers were trained on speech-silence patterns derived from the teacher utterances. The best models were able to detect Q&A segments in windows of 90 seconds with an AUC (Area Under the Receiver Operating Characteristic Curve) of 0.78 in a manner that generalizes to new classes. Implications of the findings for automatic coding of classroom discourse are discussed.

Keywords

Dialogic instruction, teacher feedback, professional development, live classrooms, speech, learning

1. INTRODUCTION

Dialogic instruction, a form of classroom discourse based around the free exchange of ideas and open-ended discussion, is considered to be an important pedagogical approach to increase student engagement [11] and improve student achievement [24]. However, the quality of implementation of dialogic instruction in classrooms varies widely. Recent research has demonstrated the importance of formative assessment of teacher use of dialogic instruction in classrooms [10]. Providing formative feedback based on what actually occurs in classrooms allows teachers to focus their efforts on improving the quality of dialogic instruction over time. Providing formative feedback efficiently, accurately, and automatically on a day-to-day basis will ensure that teachers receive the feedback they need to better incorporate dialogic instructional practices into their classrooms. However, large-scale efforts to assess the quality of classroom discourse have relied on manual, labor-intensive, and expensive excursions into classrooms. The automation of classroom discourse analysis to inform personalized formative assessment and training programs has the potential to transform teachers' use of dialogic instruction and thereby improve student outcomes. This is the overarching goal of the current project, called CLASS 5.

The CLASS 5 project is focused on automatically analyzing classroom discourse as a means of providing feedback to teachers. CLASS 5 is intended to be a modern adaptation of the traditional model of requiring trained observers to manually code classroom discourse, an unsustainable task for providing day-to-day feedback for professional development. The automated analyses are grounded in the coding scheme of Nystrand and Gamoran [6,19], who observed thousands of students across hundreds of middle and high school English Language Arts classes. They found that the overall dialogic quality of classroom discourse through teacher's use of authentic questions (questions without prescripted responses), uptake (integration of previous speaker's ideas into future questions), and classroom discussion had positive effects on student achievement. The Nystrand and Gamoran coding scheme has been validated in multiple studies across a multitude of classrooms [2,7,17,18], hence, we are optimistic that by automating this coding scheme, we will replicate the well substantiated results of finding positive effects of dialogic instruction on student achievement. In the remainder of this section, we provide a brief overview of the Nystrand and Gamoran coding scheme, review prior work on automated classroom discourse analysis, and provide a brief overview of the present study, which is focused on automatically detecting question-answer (Q&A) segments via audio recordings of teachers during normal classroom instruction.

1.1 Coding Classroom Discourse

The Nystrand and Gamoran [6,19] coding scheme can be subdivided into three key '*tracks*,' of increasingly fine granularity: 1) episodes, which refer to the activity/topic being addressed by the teacher; 2) segments, seventeen categories that represent possible techniques used to implement the episode; and 3) questions asked by teachers or students embedded within segments [19]. Each track can be further understood by its own nuance and properties. For example, many classes typically begin

and end with procedural episodes (i.e., "getting started"; "preparing to leave") with one or more instructional episodes permeating the core of the class. All episodes consist of one or more segments, which can be broadly subdivided into four categories: classroom management activities, direct instruction, seatwork, and tests and quizzes. Questions are coded along dimensions of authenticity, uptake, and cognitive level as elaborated in [19].

Our current focus is on classifying key *segments* in classroom discourse. Of the seventeen segment categories the most frequent segments are lecture (including film, music, or video), Q&A, reading aloud, supervision/helping, and small group work [19]. Lecture incorporates instances where a teacher speaks for at least 30 seconds on a topic unrelated to the procedural aspects of running a class (discussing assignment instructions, for example, would not be considered lecture). Q&A segments include a question or series of questions which are non-rhetorical, non-procedural, and non-discourse management questions. Reading aloud segments consist of students reading aloud. Supervised/helping segments occur when teachers help students complete individual work. Small group work segments occurs when a group of students participates in some activity.

Discussions constitute an important, but rare, segment of particular relevance to dialogic instruction. According to the coding scheme, discussion segments consist of a free exchange among three or more participants that lasts longer than 30 seconds. Discussions typically include relatively few questions. Questions that are asked tend to focus on clarification of ideas. Discussions are typically initialized when a student makes an observation, rather than asking a question, and another student or a teacher asks for clarification on that observation. In contrast, Q&A segments usually consist of three parts – an initiation, a response, and an evaluation (IRE). The most common example of these parts begins with a teacher question, followed by a student answer, and then a teacher response to the student's answer. The teacher's response is often perfunctory (e.g. 'right' or 'wrong') – and sometimes non-vocalized (i.e., a nod) [16,18].

Q&A and discussion segments have traditionally positively correlated with achievement, and it is recommended that teachers should attempt to maximize use of these segments [19]. As mentioned above, discussion segments are rare in classrooms. In Nystrand's observations there was on average less than one minute of discussion per class [19]. Traditionally Q&A segments have dominated between 30% - 42% of class time [19]. In fact, when discussion does occur it tends to do so in the midst of Q&A segments. Therefore, the present study focuses on the automated detection of Q&A segments as an initial approach to automating the coding of classroom discourse.

1.2 Related Work

The closest work in this area stems from research by Wang and colleagues. In particular, Wang et. al. [26] used teacher and student speech features obtained by the Language Environment Analysis system (LENA) [5] to analyze discourse profiles from 1st to 4th grade math classes. LENA is a wearable system which records and measures the quality of language produced by and directed at young children. Wang et. al. had two trained coders listen to 30-second audio windows and classify if the window represented discussion, lecture, or group work. Coders also provided their confidence in their annotation on a scale of 1 to 3 (1 indicating a lack of confidence and 3 indicating very confident).

LENA was adapted to assess when teachers were speaking, students were speaking, speech was overlapping, or there was silence. Wang et al. [25] previously found that LENA coded many student utterances as teacher utterances and modified LENA to improve its voice detection accuracy by changing the categorization algorithm to account for volume as an indicator of the distance between the speaker and the microphone. Their precision for teacher speech detection ranged from 0.95 – 0.99 and their precision for student speech detection ranged 0.70 to 0.86.

They then trained a random-forest classifier to classify the 30-second windows based on the results of speech segmentation. They used one coder's confidence labels of 3 for training data. This constituted 62% of the windows. They validated their model on all of the windows (including the training windows), but with the annotations provided by a different coder. The coders agreed on 83% (Kappa 0.72) of the annotations, so there was considerable overlap between training and testing data. Their model achieved an accuracy of 83% (Kappa of 0.73) in discriminating between lecturing, discussion, and group work.

Although Wang et. al. [26] reported success at classifying classroom discourse at course-grained levels, their audio solution was focused on what occurred in the context of individual windows, rather than using the broader classroom context to code segments. Further, according to Wang's coding, discussion occurred approximately 33% of the time, indicating their definition of discussion was much more inclusive than the Nystrand & Gamoran coding scheme [6,19]. Their definition of discussion, which involved students and teachers having conversations about the learning content on the whole class level (the conversation should be accessible to the majority of students in class), is not incorrect, but more closely aligns with our definition of Q&A segments. In addition, their validation method did not include an independent class-level hold-out set, thus evidence for generalizability to new classes is unclear.

1.3 Current Study

The present study takes inspiration from Wang et al.'s pioneering work, but also differs from it in significant ways. The LENA system is a research-grade solution and is thereby cost prohibitive and might not be scalable. This raises the question of whether classroom discourse can be automatically analyzed using more cost effective consumer-grade sensors. Of particular interest is addressing which signals are needed for accurate automatic classification of classroom discourse. Teachers lead dialogic instruction and one possibility is the only signals needed to capture classroom activity are signals that capture teacher activity. Since teachers may be anywhere in a classroom, data needs to be collected from a device that accompanies their movements with high fidelity. One attractive candidate for such a sensor is a microphone to record teacher speech, which is the approach adopted here.

Recording teacher speech is not a difficult task, but distilling the signal into appropriate features for classification of Q&A segments is more complicated. Thus, we first focused our efforts on teacher utterance detection in an attempt to find the onsets and offsets of teacher speech. Features extracted from these onsets and offsets, signaling periods of speech and rest, were then used to train classifiers to discriminate Q&A segments from all other

segments combined (i.e., Q&A vs. "other" discriminations). Note that all classification is done by analyzing these utterance onsets and offsets in an attempt to establish the accuracy of Q&A segment classification using a minimalistic approach.

The key differences between the present approach and Wang's previous work include: (a) our use of a consumer-grade microphone rather than the LENA system; (b) segments are coded during live classrooms, so that the overarching classroom context can be incorporated in the coding; (c) we study Q&A segment classification by exclusively focusing on the teacher speech signal; and (d) our models are validated across class sessions, thereby ensuring generalizability to new classes.

The remainder of the paper is organized as follows. First, we discuss our data collection, which involved coders trained in Nystrand's coding scheme collecting data from three teachers in 21 class sessions over the course of a semester (Section 2). We recorded teacher speech using a headset microphone and the audio signal was temporally synchronized with the human codings. Next, we developed an amplitude envelop-based utterance detection approach to segment the teacher audio into periods of speech and rest (Section 3). Then, supervised classifiers were used to detect Q&A segments from features extracted by the utterance detection algorithm (Section 4). Implications of our findings towards the broader goal of automating the analysis of classroom discourse at multiple-levels are discussed (Section 5).

2. Data Collection

Audio recordings were collected at a rural Wisconsin middle school during literature, language arts, and civics classes. The recordings were of three different teachers: two males – Speaker 1 and Speaker 2 – and one female – Speaker 3. The recordings spanned classes of about 45 minutes each on 9 separate days over a period of 3-4 months. Due to the occasional missed session, classroom change, or technical problem, a total of 21 classroom recordings were available for analyses. During each class session, teachers wore a Samson AirLine 77 'True Diversity' UHF wireless headset microphone that recorded their speech, with the headset hardware gain adjusted to maximum. This microphone was chosen for its high noise-cancelling ability and is not cost-prohibitive ($300 per unit). Audio files were saved in 16 kHz, 16-bit mono .wav format. Teachers were recorded naturalistically while they taught their class as usual.

Two observers trained in Nystrand et. al.'s dialogic coding technique [19,20] were present in the classroom during recordings. Observers used a specialized coding software developed by Nystrand [15] to mark episodes, segments, and teacher's dialogic questions with the appropriate labels, as well as start and stop times as the class progressed. Later, these same observers reviewed the recordings to ensure labels were accurate and engaged in discussion until all discrepancies were resolved.

Table 1 lists the proportion of time spent on each of the segments. We note that Q&A segments were the most frequent, while discussions were highly infrequent. Other somewhat frequent segments include small group work, supervised/helping, and lecture/film/video/music. The subsequent analyses focus on detecting the 28.6% Q&A segments from all other segments combined.

Table 1. Proportion of class time on each segment

Segment	Proportion
Question/answer	**0.286**
Small Group Work	0.160
Supervised/helping	0.158
Lecture/film/video/music	0.150
Reading Aloud	0.093
Procedures and directions	0.091
Supervised/monitoring	0.019
Silent Reading	0.017
Other	0.012
Unsupervised seatwork	0.006
Class interruption	0.003
Game	0.002
Discussion	0.001

3. TEACHER UTTERANCE DETECTION

Our overall objective was to use teacher speech to detect instances of question-and-answer using recorded audio from classrooms. Before this could be done, recorded audio needed to be distilled into instances of teacher speech vs. rest (silence or no speech). Thus, we developed and validated an utterance detection method as discussed below.

3.1 Method

Our first assumption was that all sound was voice because teacher speech was recorded from a high-quality noise-canceling headset microphone, all sound was voice and that no advanced voice activity detection (VAD) techniques were required[1]. Thus, a simple binary procedure was used for utterance detection. The amplitude envelope of the teacher's low-pass filtered speech was passed through a threshold function in 20 millisecond increments. Where the amplitude envelope was above threshold, the teacher was considered to be speaking. Where the amplitude envelope was below threshold, the teacher was assumed to not be speaking. Any time speech was detected, that speech was considered part of an utterance, meaning there was no minimum threshold for how short an utterance could be. Utterances were marked as complete when speech stopped for 1000 milliseconds (1 second). A typical result of this automatic utterance labeling method is depicted in Figure 1.

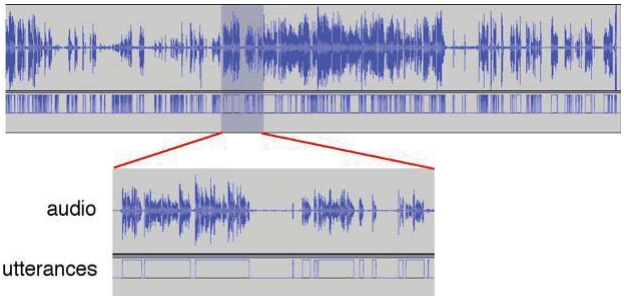


Figure 1: A 45-minute class recording (top) is depicted, while a small portion of the recording is enlarged for a detailed view (bottom). The upper track visualizes the .wav form of the audio. The lower track visualizes detected utterances.

[1] We also experimented with off-the-shelf voice activity detection algorithms [22], with comparable, if not slightly inferior, results.

The speech delimiter and threshold were both low to ensure all speech was detected, resulting in no known cases of missed speech. This process resulted in 8662 utterances, which we call *potential speech utterances*. An examination of a subset of these potential speech utterances indicated that there were a large number of false alarms. These were mainly attributed to instances of background noise permeating the audio. Common examples of background noise that the microphone picked up included voices of students who were being exceptionally loud, sounds from a film or audio clip being played in the classroom, and sounds of the teacher's breathing.

A two-step filtering approach was taken to eliminate the false alarms. First, potential utterances less than 125 milliseconds in length (12% in all) were deemed to be too short to contain meaningful speech and were eliminated. Second, the remaining potential speech utterances were submitted through an automatic speech recognizer (Bing Speech) in an effort to identify the false alarms. Bing Speech [13] is a freely available, cloud-based automatic speech recognition service which supports seven languages. Bing returns a recognition result and a confidence score for that speech. Instances where Bing rejected the speech or where it returned no transcribed text were considered to be false alarms. After eliminating the false alarms, we were left with a total of 5502 utterance (64% of the 8662 potential utterances).

3.2 Validation

A small study was conducted to evaluate the aforementioned utterance detection method. A random sample of 500 potential utterances was selected and manually annotated for speech/non-speech. Speech was defined to include all articulations (i.e., "um", "hm", "sh", etc) in addition to normal spoken segments. Potential speech utterances that included noise (i.e., loud students) in addition to teacher speech, the utterance was deemed as being a spoken utterance since it contained teacher speech. In total, 63% of potential utterances contained teacher speech and 37% did not. Thus, the effective false alarm rate prior to discarding utterances less than 125 milliseconds in length and accepted by Bing Speech was 37%.

Table 2 presents the confusion matrix obtained when using the 125 millisecond utterance duration threshold and Bing Speech to eliminate false alarms in the sample of 500 potential utterances. The filtering approach was highly successful, resulting in a kappa of 0.93 (agreement between computer-detected teacher utterances and human-detected teacher utterances). We note a substantially high hit and correct rejection rates and very low false alarms and miss rates. This was deemed to be sufficiently accurate for the present goal of detecting Q&A segments from teacher speech.

Table 2. Descriptive Statistics of Utterances

	Predicted	
Actual	*Speech*	*Non-Speech*
Speech	0.96 (hit)	0.04 (false alarm)
Non-Speech	0.03 (miss)	0.97 (correct rejection)

4. CLASSIFYING Q&A SEGMENTS

Segments were coded in the classrooms of three teachers in 21 classes by trained coders over the course of a semester. Our goal was to differentiate Q&A segments, which are key for dialogic instruction, from all other types of segments (a binary Q&A segment vs. "other" classification task). Features for Q&A segment classification were obtained from the automated teacher speech utterance detection approach discussed above.

4.1 Method

4.1.1 Creating and labeling instances

Audio was sectioned into non-overlapping windows of 30, 45, 60, 75, and 90 seconds in length. Each window was assigned a label of "Q&A" or "other" based on the annotations by the trained coders (see Section 2). In some cases, there was overlap, defined as a window with multiple segment labels (e.g., first 20 seconds are Q&A and the last 10 seconds are lecture). For windows with overlap, the label of "Q&A" or "other" was assigned based on the label of the majority segment (e.g., Q&A in the example above).

Table 3 presents the number of windows and the proportion of windows that contain overlap for each window size. As expected, the proportion of windows with overlap increases as window size is increased.

Table 3. Number of instances and proportion of instances with overlap

Window	**N**	**N (with overlap)**	**Proportion with overlap**
30 seconds	1886	163	0.09
45 seconds	1253	145	0.12
60 seconds	937	126	0.13
75 seconds	748	126	0.17
90 seconds	620	112	0.18

Note: N = Total number of windows in a dataset

4.1.2 Feature Engineering

Features were based on teacher utterance detection as discussed in Section 3. The features attempt to capture the temporal speech patterns that teachers use in Q&A segments as defined by the initiation (speech), response (rest), and evaluation (speech) pattern of Q&A discussed in Section 1.1. They include: 1) number of utterances, 2) mean utterance duration, 3) standard deviation of utterance duration 4) minimum utterance duration 5) maximum utterance duration, 6) number of rests, 7) mean rest duration (rests were the intervals of silence between utterances), 8) standard deviation of rest duration, 9) minimum rest duration, 10) maximum rest duration, and 11) window number, the number of windows into a class session.

4.1.3 Model Building

Supervised classifiers were built using the Waikato Environment for Knowledge Analysis (WEKA) [9] an open source data mining tool. Models were cross validated on the class level to ensure generalizability across class sessions. In each fold, a random 67% of the classes were used for training and the remaining 33% were used for testing. This process was repeated for 25 iterations and the classification accuracy metrics was averaged across these iterations. A large number (N = 43) of standard classifiers were tested because of a lack of knowledge regarding what classifier works best for this type of data.

Various data treatments were applied in order to determine which combination resulted in the best model. First, tolerance analysis was used to eliminate features that exhibited multicollinearity [1]. Second, four feature selection algorithms: 1) Information Gain Ratio (Info-Gain) [14], 2) RELIEF-F [12], 3) Gain-Ratio [21], and 4) Correlation-based Feature Selection (CFS) [8] were used

(on training data only) to select either 25%, 50%, or 75% of the top features (the specific percentage of features was another parameter). Third, the data was Winsorized by setting outliers greater than 3 standard deviations from the mean to the corresponding value 3 standard deviations from the mean. Finally, synthetic minority oversampling technique (SMOTE) [4] was applied to the training data by creating synthetic instances of the minority Q&A class until the classes were balanced. Testing data was not sampled.

4.2 Results

4.2.1 Best Models

Classification accuracy was evaluated with area under the receiver operating characteristic curve (AUC), a metric bounded on [0, 1] with 1 indicating perfect classification and 0.5 indicating chance level classification. Table 4 presents an overview of the AUCs associated with the best models for each window size. The mean AUC across all windows was 0.73 (SD = 0.05). Classification accuracy was greater for longer window sizes with the best results obtained for the 90 second window. This model used a logistic regression classifier and had 5 features (discussed below). Table 5 presents the confusion matrix for this 90 second window model. The main source of errors appear to be misses rather than false alarms.

Table 4. AUC for best models at each window size

Window Size	AUC
30 secs	0.67 (0.04)
45 secs	0.69 (0.05)
60 secs	0.75 (0.04)
75 secs	0.75 (0.04)
90 secs	0.78 (0.05)

Note: Standard Deviation in parenthesis

Table 5. Confusion matrix for best model using class-level cross-validation

	Predicted		
Actual	*Q&A*	*Other*	**Priors**
Q&A	0.78 (hit)	0.22 (false alarm)	0.26
Other	0.36 (miss)	0.64 (correct rejection)	0.74

4.2.2 Robustness to Overlap

One concern was whether classification accuracy was degraded due to instances where Q&A segments overlapped other segments within a window. As presented in Section 4.1, the larger the window size, the greater proportion of instances that contain overlap. To study the effect of overlap, we built another set of models with overlapping segments removed.

Performance of models without overlapping windows was consistent compared to models with overlapping windows (see Table 4). Mean AUC for the models built without overlap was 0.74 (SD = 0.04) compared with mean AUC from Section 4.2.1: 0.73 (SD = 0.05). Thus, our best models were robust to instances where Q&A segments overlapped with other segments within a window.

4.2.3 Feature Analysis

We analyzed the five features used in the best model (90 second window). These features were 1) number of utterances, 2) mean utterance duration, 3) maximum utterance duration, 4) mean rest duration, 5) maximum rest duration. Table 6 presents the mean and standard deviation for these top features across the four most frequent segments (see Table 1). All non-Q&A segments included a fewer number of utterances, shorter utterance durations, and fewer silences (rest). For lecture/media this was likely a result of the all-inclusiveness nature of lecture/media which could include instances of only speech, a traditional lecture, or instances of no speech (e.g., when a film is played). For group work, this was likely because speech consisted of clarifying instructions or addressing individual group concerns. Supervised/helping was likely similar to group work, but rather than group concerns, individual concerns were addressed.

Table 6. Mean and standard deviation for features across most frequent segments

Feature	Q&A	Lecture/ Media	Small Group Work	Supervised/ Helping
Number of utterances	10.45 (4.82)	4.86 (5.16)	8.90 (4.32)	7.38 (4.46)
Mean utterance duration	5.19 (4.15)	3.23 (4.37)	2.76 (1.83)	2.80 (1.92)
Maximum utterance duration	14.62 (9.85)	7.77 (9.44)	7.80 (5.69)	8.14 (7.02)
Mean rest duration	5.40 (4.67)	38.71 (37.26)	12.22 (19.23)	17.57 (24.77)
Max rest duration	15.92 (11.71)	50.42 (33.53)	27.91 (22.31)	35.51 (25.60)

Note: Standard Deviation in parenthesis

5. General Discussion

Dialogic instruction is considered to be an important pedagogical approach for promoting learning and engagement in classrooms. However, analyzing the effective use of dialogic instruction in classrooms has traditionally required the presence of trained live coders and is inherently non-scalable. In the present paper, we considered the possibility of automating the coding of classroom discourse. As an initial step, we focused on automatically detecting question-and-answer (Q&A) segments, an important component of dialogic instruction, using teacher speech. We were able to detect instances of Q&A from teacher speech with moderate success in live classrooms. In this section, we compare our results to previous work in this area, discuss major findings, limitations of the present study, and consider next steps with this research.

5.1 Comparing with Previous Work

Our goal was to compare our approach, which only uses features from teacher speech, with models from Wang et al. [26], which were based on teacher speech, student speech, overlapping speech, and silence. A perfect comparison is complicated due to many differences across approaches, most importantly with

respect to how classroom activities were coded and how the models were validated. In particular, coders in the Wang et al. study annotated their data using 30-second intervals and specified a confidence level for each annotation. This allowed them to train their models on only the high-confidence labels. In comparison, we used a variety of different window sizes and our labels did not include a confidence level.

Our best model, which used a logistic regression classifier, had a kappa of 0.32, which is much lower than Wang et al.'s kappa of 0.77. To equate models, we also experimented with using a random forest model [3], used by Wang et al. Using a random forest model and validating at the class-level resulted in an AUC of 0.71 (SD = 0.04) and a lower kappa of 0.25 (SD = 0.07). However, we noted that Wang et al. validated their data using both training and testing data, while our models were validated on held-out class sessions. In other words, 62% of their testing data contained training instances. We attempted to replicate their validation approach by randomly selecting 62% of training instances for inclusion in the testing data. This drastically increased the AUC to 0.87, with a Kappa of 0.57.

In conclusion, although our model's performance is lower than Wang et al's, there are many possible reasons for this difference. For example, differences in our definitions of Q&A, their coding of each window devoid of context (which could lead to misinterpreting a window due to lack overall of context), different recording setups (LENA vs. microphone), different class structures (elementary mathematics vs. middle-school literature, language arts, and civics classes), and so on. Future work needs to equate these differences so the two approaches can be compared more equitably.

5.2 Major Findings

We were moderately successful in detecting Q&A segments despite considerable challenges associated with automatically recording classroom discourse using only teacher speech recorded via a headset microphone. Our major contribution is the use of consumer grade equipment to filter teacher utterances from non-teacher utterances in a noisy classroom environment. We found that we could use those utterances to develop and validate Q&A segment detectors in classrooms using only teacher speech.

Our approach consisted of two steps. Step 1 involved segmenting teacher utterances and Step 2 involved analyzing speech-silence dynamics from this segmentation to train classifiers suitable for discriminating Q&A segments from all other coded segments. For utterance detection, we used an amplitude enveloping approach to identify a large subset of potential teacher utterances and filtered them based on both duration and by submitting them to a web-based automatic speech recognizer (Bing Speech). We validated the utterance detection approach using a sample of 500 potential speech utterances randomly sampled from three teachers and 21 class sessions. We reliably and accurately discriminated speech from non-speech (kappa of 0.93) and this was accomplished despite the complexities of teacher utterance detection in noisy classrooms such as loud student speech, classroom disruptions, the use of media (i.e., video, music), and non-articulations of the teacher (such as breathing).

For Step 2, we built models to classify instances of Q&A from other instructional activities using speech-silence dynamics from the utterance segmentation. The best model was a logistic regression classifier trained on speech and silence features in 90 second windows which yielded an AUC of 0.78 when validated at the class-level. We also built models without overlap in order to determine their effect. The models without overlap were equitable to models with overlap, indicating our models were robust to this issue. Finally, we analyzed the top features from our best model and the main finding was that Q&A segments were associated with more teacher speech and fewer rests compared to the other segments.

5.3 Limitations and Future Work

This study was not without its limitations. First, data was collected from three teachers who taught different subjects. However, this is a small number of teachers and all taught at the same school, so replication with a larger and more diverse sample is warranted. Second, discussion is a key indicator of dialogic discourse in classrooms [19], but our data set had only one instance of discussion, which lasted 77 seconds. Thus models could not be built for this key activity. Finally, our method focuses on a coarse-grained measure of classroom discourse. Future research is needed before a fine-grained analysis of the types of questions being asked in Q&A segments can be done (see Samei et al. [23]). When we use Bing to filter speech, it returns recognition results which could potentially be used for these fine-grained analysis. This is an important item for future work.

In general, future data collection should include more teachers, schools, social environments, and class diversity. Future work should also consider ways to capture student speech in an equally cost effective way. One possibility would be to record the entire room with a boundary microphone. However, it should be noted that every additional sensor increases the complexity of data collection and raises the threshold of adaptation in terms of cost and complexity of use. For example, if using a boundary microphone to capture student speech, a teacher needs to learn where best to position the microphone. However, a headset microphone only requires a teacher to turn it on and wear it. Nevertheless, we anticipate much improved results in Q&A detection when student speech is available.

5.4 Concluding Remarks

The overall purpose of this research was to automate the coding of classroom discourse and the present paper made some advances in this direction. As Nystrand et al. found [19], professional development activities focused on increasing the quality of dialogic instruction can have measurable effects on student achievement. The automated classroom discourse analysis techniques developed here can contribute to this goal by providing daily feedback to teachers for their professional development. Although this feedback alone may allow teachers to better reflect on their classroom instruction, it remains to be seen whether this increases their use of appropriate techniques for dialogic instruction. If not, tracking key components of dialogic instruction allows for interventions to increase dialogic instruction in classrooms. The research presented here represents an important initial step toward these goals, the next step involving an analysis of individual question-events at a more fine-grained level.

6. ACKNOWLEDGMENTS

We would like to thank Dr. Michael Brady for the amplitude envelope processing method.

This research was supported by the Institute of Education Sciences (IES) (R305A130030). Any opinions, findings and

conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the IES.

7. REFERENCES

1. Allison, P.D. *Multiple regression: A primer.* Pine Forge Press, 1999.
2. Applebee, A.N., Langer, J.A., Nystrand, M., and Gamoran, A. Discussion-Based Approaches to Developing Understanding: Classroom Instruction and Student Performance in Middle and High School English. *American Educational Research Journal 40*, 3 (2003), 685–730.
3. Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, (2011).
5. Ford, M., Baer, C.T., Xu, D., Yapanel, U., and Gray, S. *The LENA Language Environment Analysis System.* Technical Report LTR-03-2. Boulder, CO: LENA Foundation, 2008.
6. Gamoran, A. and Kelly, S. Tracking, instruction, and unequal literacy in secondary school English. *Stability and change in American education: Structure, process, and outcomes*, (2003), 109–126.
7. Gamoran, A. and Nystrand, M. Background and instructional effects on achievement in eighth-grade English and social studies. *Journal of Research on Adolescence 1*, 3 (1991), 277–300.
8. Hall, M.A. *Correlation-based Feature Selection for Machine Learning.* 1999.
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl. 11*, 1 (2009), 10–18.
10. Juzwik, M.M., Borsheim-Black, C., Caughlan, S., and Heintz, A. *Inspiring Dialogue: Talking to Learn in the English Classroom.* Teachers College Press, 2013.
11. Kelly, S. Classroom discourse and the distribution of student engagement. *Social Psychology of Education 10*, 3 (2007), 331–352.
12. Kononenko, I. Estimating attributes: analysis and extensions of RELIEF. *Machine Learning: ECML-94*, Springer (1994), 171–182.
13. Microsoft. *The Bing Speech Recognition Control.* 2014. http://www.bing.com/dev/en-us/speech. Accessed 14 Jan 2015
14. Mitchell, T.M. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill 45*, (1997).
15. Nystrand, M. *CLASS (Classroom language assessment system) 2.0: A Windows laptop computer system for the inclass analysis of classroom discourse.* Wisconsin Center for Education Research, Madison, 1988.
16. Nystrand, M. CLASS 4.0 user's manual. *The National Research Center on*, (2004).
17. Nystrand, M. Research on the Role of Classroom Discourse as It Affects Reading Comprehension. *Research in the Teaching of English 40*, 4 (2006), 392–412.
18. Nystrand, M. and Gamoran, A. Instructional discourse, student engagement, and literature achievement. *Research in the Teaching of English*, (1991), 261–290.
19. Nystrand, M., Gamoran, A., Kachur, R., and Prendergast, C. Opening dialogue. *Teachers College, Columbia University, New York and London*, (1997).
20. Nystrand, M., Wu, L.L., Gamoran, A., Zeiser, S., and Long, D.A. Questions in time: Investigating the structure and dynamics of unfolding classroom discourse. *Discourse processes 35*, 2 (2003), 135–198.
21. Quinlan, J.R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.
22. Rouvier, M., Dupuy, G., Gay, P., Khoury, E., Merlin, T., and Meignier, S. *An open-source state-of-the-art toolbox for broadcast news diarization.* Idiap, 2013.
23. Samei, B, Olney, A, Kelly, S, et al. Domain independent assessment of dialogic properties of classroom discourse. *Proceedings of the 7th International Conference on Educational Data Mining*, Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (2014), 233–236.
24. Sweigart, W. Classroom Talk, Knowledge Development, and Writing. *Research in the Teaching of English 25*, 4 (1991), 469–496.
25. Wang, Z., Miller, K., and Cortina, K. *Using the LENA in Teacher Training: Promoting Student Involement through automated feedback.* na, 2013.
26. Wang, Z., Pan, X., Miller, K.F., and Cortina, K.S. Automatic classification of activities in classroom discourse. *Computers & Education 78*, (2014), 115–123.

Topic Transition in Educational Videos Using Visually Salient Words

Ankit Gandhi*
Xerox Research Centre India
Ankit.Gandhi@xerox.com

Arijit Biswas*
Xerox Research Centre India
Arijit.Biswas@xerox.com

Om Deshmukh
Xerox Research Centre India
Om.Deshmukh@xerox.com

ABSTRACT

In this paper, we propose a visual saliency algorithm for automatically finding the topic transition points in an educational video. First, we propose a method for assigning a saliency score to each word extracted from an educational video. We design several mid-level features that are indicative of visual saliency. The optimal feature combination strategy is learnt from a Rank-SVM to obtain an overall visual saliency score for all the words. Second, we use these words and their saliency scores to find the probability of a slide being a topic transition slide. On a test set of 10 instructional videos (12 hours), the F-score of the proposed algorithm in retrieving topic-transition slides is 0.17 higher than that of Latent Dirichlet Allocation (LDA)-based methods. The proposed algorithm enables demarcation of an instructional video along the lines of 'table of content'/'sections' for a written document and has applications in efficient video navigation, indexing, search and summarization. User studies also demonstrate statistically significant improvement in across-topic navigation using the proposed algorithm.

Keywords

visual word saliency, ranking, topic transition, educational videos, video demarcation and indexing

1. INTRODUCTION

The rapid growth of online courses and Open Educational Resources (OER) is considered to be one of the biggest turning points in education technology in the last few decades. Many top-ranked universities and educational organizations across the world are making thousands of video lectures available online for no cost either in the form of Massively Open Online Courses (MOOCs) or as open access material. A few national governments have also formulated policies to record classroom lectures from top-tier colleges and make them freely available online (e.g., National Program of Technology Enhanced Learning (NPTEL)[1] in India). This online content can either assist classroom teaching in educational institutions with limited resources or aid out-of-class learning by the students.

As the amount of this online material is increasing rapidly (tens of thousands of hours of video currently), it is important to develop methods for efficient consumption of this multimedia content. Developing methods for summarization [2, 3], navigation [4] and topic transition[5, 6, 7, 8], for educational videos are now active areas of research.

One of the most challenging areas of research is to automatically identify time instances where a particular topic ends and a new one beings (i.e., topic transitions) in an educational video. Consider this real-classroom example: Professors often teach multiple topics within a lecture (of, say, 60-75 minutes). For example, in a lecture video[1] on support vector machine (SVM), the professor might cover the definition of version space, motivation for SVM, primal formulation, dual formulation, support vectors and perhaps end the lecture with kernel formulation. When a student is viewing this video lecture s/he might only be interested in the part where the professor is discussing, say, the dual formulation for SVM. This frequently happens when only a few topics of the video are relevant for the student or when the student wants to revise particular concepts for an upcoming assessment. In such a situation the student would typically 'guesstimate' the location with multiple back and forth navigations of the video. [Indeed, in a large-scale study on the EdX platform, authors in [9] found that certificate earning students, on an average, spend only about 4.4 minutes on a 12-15 minute-long video and skip about 22% of the content.] Finding these topic transition points in long videos can be extremely difficult and time-consuming. On the other hand, if the lecture videos can be automatically annotated with the locations where the topic is changing (e.g., dual formulation start point, primal formulation start point, etc.), the student can easily navigate through these locations and find the topics of interest efficiently.

A human expert familiar with the topic of a lecture can manually go through each lecture video and label the topic transition points. However as the quantity of online video lectures increases, manually labelling topic transition points for all of them is going to be a highly time consuming and expensive process. Demarcating these topic transitions is straightforward in written documents as the authors tend to

*Equal contribution.

[1]https://www.youtube.com/watch?v=eHsErlPJWUU

create table of contents or sections and subsections. Video lectures, by the very nature of the medium, don't have such demarcation. It is the goal of this research work to automatically identify these topic transitions in educational videos and highlight these 'sections' to the end user.

In this paper, we propose a novel approach where the visual content of a lecture video is analyzed to determine the transition points. In the proposed approach, the visually salient or important words are extracted from the frames of an educational video and these words along with their saliency scores are used to identify the points where the topic is changing in the video. Two major novel contributions of this work are:

1. **Visual saliency of words:** Since we use the visual content in an educational video to find out the topic transition points, one major challenge was to figure out the visual cues that are most important for determining the transition points. Intuitively it is clear that the words used in the slide frames [2] and their distribution can be used to determine the change of topics. However we also figured out that how a word is used in a particular slide provides significant cues regarding the word's significance in topic transition. For example, if a word is bold and located towards the top or left of the page, they contribute more in the topic transition than words which are located at the bottom right corner of a slide. An underlined word is usually more important than other words in a slide frame. To capture these visual characteristics, we propose seven novel mid-level features for the words present in educational videos. These features are called *underlineness, boldness, size, capitalization, isolation, padding,* and *location*. Once we extract all of these features for a word they are combined using a weight vector to create a saliency score corresponding to every word in the video. To learn this optimal weight vector we propose a novel formulation of the Rank-SVM algorithm [10] on human-annotated salient words (described in Section 4).

2. **Topic transition:** Once we extract the words and their corresponding saliency scores from a video, the next step is to find the topic change points. The saliency scores are used to estimate (a) how many novel yet salient words are introduced in each slide (referred to as Salient-Word-Novelty), and (b) number of lower saliency words in earlier slides that occur with higher saliency (referred to as Relative Saliency), for a particular slide. We propose novel methods for visual content-based across-slide computation of these two features for every slide and formulate a posterior model to estimate the probability that a given slide is a topic transition slide.

Note that the proposed approach is applicable for educational videos where slides are fully or at least partially used as word recognition accuracy for hand-written text in images is extremely poor and still an open research problem. We observed that a sizable majority of the OER is based on slideware.

[2]Throughout the paper by slide frame/slide we mean the frames of an education video where the teacher is displaying a slide. We also assume that the power point (.ppt) slide file is not separately available along with the video.

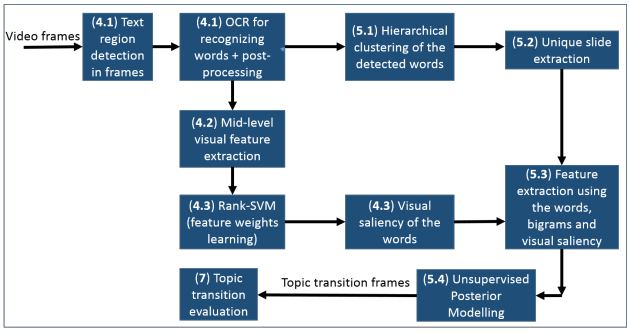


Figure 1: *Pipeline of the proposed system. Figure also shows the corresponding section numbers where details of each component are explained.*

The performance of the proposed approach in identifying topic transition locations was evaluated on 10 different lecture videos with a total duration of 12 hours chosen from the NPTEL set. The proposed approach outperforms the topic transition points derived using the well-known topic modelling approach [11] by an F-score of 0.17 (0.6 to 0.77 where the maximum possible F-score is 1). User studies demonstrate statistically significant improvement in across-topic navigation using the proposed algorithm.

2. RELATED WORK

Topic segmentation of instructional videos is an active area of research. All the work however focuses on analysing the filming aspects of the video and not the educational content.

Authors in [5] proposed a method for high level segmentation of topics in an instructional video using the variation in the content density function. The key contributing factors which manipulate the content density function are shot length, motion and sound energy. This work is extended in [6], where a thematic function is introduced to capture the frequency of appearance of the narrator, frequency of the superimposed text and narrator's voice over. The thematic function is used along with the content density function in a two tiered hierarchical algorithm for segmenting the topics. The authors in [7] propose hidden markov model (HMM) based approaches for topic transition detection. First audio-visual features are extracted from shots in a video and each shot is classified into one of the five classes: direct-narration, assisted-narration, voice-over, expressive-linkage and functional linkage. Direct-narration/assisted-narration/voice-over implies segments where the narrator is seen in the video or not. Functional linkage is captured by large superimposed text or music playing in background. Expressive linkage is used to create the mood for the subject being presented, e.g., houses with fire images in fire safety videos. Then a two level HMM is trained using a training dataset and topic transition points are found out.

All of these approaches were developed mainly for videos used in industries to train people and to convey instructions and practices, e.g., fire safety video. However OER videos, where the teacher goes over the content of slides, are very different from these kinds of videos. The camera captures the teacher and the content interchangeably with the content being more on focus. OER videos do not have music playing in background, images for mood creation, variation in sound energy or significant amount of motion. Thus all

of these prior methods will not be applicable for the educational videos of our interest. More importantly, none of these methods capture the actual content or their characteristics like saliency to model the topic change.

The proposed solution for topic transition will also drive other applications related to educational videos such as non-linear navigation [4] and summarization [2, 3] which are also active areas of research.

3. SYSTEM OVERVIEW

A pipeline of the proposed system is shown in Figure 1. In the next two sections (Section 4 and Section 5), we describe the technical detail of each of the components shown in the figure. The input to the system is uniformly sampled frames extracted from an educational video.

4. VISUAL SALIENCY

In this section, we discuss the steps involved in assigning visual saliency scores to words present in slides.

4.1 Word Recognition and Text Post-processing

The first step of our pipeline is to recognize words in frames from an educational video. Recognizing text from images [12] is an extremely hard problem and continues to be an active area of research in computer vision/image processing. Words recognition usually involves two steps, first, localization of text in the frame, and then identification of text in the localized regions. In our proposed approach, we have used the algorithm proposed by Neumann *et al* [13] for localizing text in frames and the open source OCR engine Tesseract [14] to identify or recognize the words in the localized regions. The recognized words and their corresponding locations will serve as the input to the next part of our system. We perform stop words removal and words stemming as a text post-processing step on the recognized words. Stop words ('and', 'it', 'the', etc.)[15] do not contribute towards the context or topic of the document. Thus removing them reduces the complexity of system without affecting any downstream processing. Also, all words are stemmed to obtain their base or root form (e.g., stemming the words 'played', 'playing', 'player' to 'play') to further reduce the complexity.

4.2 Saliency Feature Computation

In this step, we compute the visual features of words that helps in determining their saliency. For computing visual features, OCR outputs, i.e., the recognized words and their locations (bounding boxes) are used. Based upon the analysis of several educational videos (different from the ones used in experiments) taken from NPTEL and edX, we formulated several visual features such as location, boldness, underlineness, capitalization, isolation, padding and size, that are indicative of visual saliency. In this section, we provide a way to quantize them and in the next section, a formal framework is proposed that combines them to predict the overall visual saliency of a word. The visual feature extraction procedure for each of the words is described below:

- **Location feature (u_1):** This feature captures the location information of a word in a slide. Generally, words which are located towards the top and left of a page are more important than the words located at the bottom and right corner of a page. We use two one dimensional Gaussian distributions ($f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$) to compute this feature. The mean of the first Gaussian distribution is set to be the left most point of an image (giving maximum score to left-most words) and the mean of the second Gaussian distribution is set to be the top most point of an image (giving maximum score to the top most words). The variance is chosen as 0.25 times the width of image and 0.16 times the height of image respectively for the two Gaussian distributions. These parameters are selected using a small validation set. For each word, top-left corner(X-Y coordinate) of its bounding box is chosen as variables in the Gaussian distributions. The location feature is given by the product of the scores obtained from the two Gaussian distributions. If a word moves away from the top left corner of an image, the location feature value gradually decreases.

- **Boldness feature (u_2):** It is usually true that if a word in a slide is relatively bolder than other words in the slide it is an important word. For computing boldness feature, first the word image is binarized. Then, the number of pixels which are foreground (i.e., the pixels which are part of the written text) are found. The pixel count is normalized with the number of characters present in the word to obtain the boldness feature. Thus, the boldness feature captures the average number of pixels occupied per character in a word.

- **Underlineness feature (u_3):** A word is underlined in a slide if the teacher wants to highlight that particular word. In this work, we use Hough Transform [16] of an image to detect line segments present in that image. Since we are only interested in horizontal or near-horizontal line segments, all other line segments are removed from consideration. We use another post-processing step to remove all the horizontal line segments which are too close to the margin. Then, all the words which are immediately above the remaining horizontal/near-horizontal line segments are assigned a non-zero score for the underlineness feature. Note that the underlineness feature for a word is binary denoting whether an underline is present below the word or not.

- **Capitalization feature (u_4):** If all the characters of a word are in upper case, then a word is assigned a non-zero score for the capitalization feature. This feature is also binary.

- **Isolation feature (u_5):** The isolation feature represents how isolated a word is in the slide. The hypothesis is that fewer the number of words in a slide, the more important the words present in it and similarly, the fewer the number of words in a line of a slide, more important the words in that line. For example, often in title slides only a title word or a phrase is present in the center of the slide. And, the title word instances are more important than their corresponding instances elsewhere. Suppose, a word w is present in line l of a slide, then the isolation feature for word w is computed as follows -

$$u_5(w) = \frac{1}{\text{No. of lines in a slide} \times \text{No. of words in line } l}$$

- **Padding feature (u_6):** In educational slides teachers often end a concept and start talking about another concept starting at the same slide. In those cases, they tend to keep usually more space before or after the title line of the new concept. We introduce a novel feature called padding to capture that information. For a word, padding feature is computed as the amount of empty space available below and above the line in which the word is present. Free space above is computed as number of pixels present between the current line and the previous line. Similarly, free space below is computed as the number of pixels present between the considered line and the next line. The sum is then normalized by the height of the image (slide) and the average line gap in the slide.

- **Size feature (u_7):** This feature captures the size of word in the slide. Words appearing with larger font are generally more important than the words appearing appearing with relatively smaller fonts. We denote the size of a word (size feature) as the height of the smallest character present in that word.

We normalize each of the visual features using 0-1 normalization across the entire video. The weighted sum of the normalized scores represents overall saliency of the words in frame. The weights are obtained using Rank-SVM[10], which we describe in the next subsection.

4.3 Learning to Rank Using Rank-SVM

In this subsection, we learn the relative importance of the visual features to predict the overall saliency of words. The weights determine how much each visual feature contributes to the overall saliency of a word. The weights were learnt by collecting a training dataset from 10 users over 5 videos. 10 slides were randomly selected from each video (hence, total of 50 slides) to collect the training set. Each slide has been shown to 3 users and thus, a single user provides data for 15 unique slides. For each slide, the user was asked the following question - "What are the salient words present in that slide that describe the overall content of the slide?". Generally, the number of salient words per slide vary between 2-12 depending upon the user and the slide. To overcome inter-user subjectivity, a word is accepted as salient only if it is marked as salient by atleast 2 users. Since in each slide users considered the selected words more salient than the words which were not selected, we can consider them as pairwise preferences. These pairwise preferences can be used in a Rank-SVM framework to learn the corresponding feature weights.

Let $\boldsymbol{u} = [u_1 u_2 \dots u_7]$ denote the visual saliency feature vector and $\boldsymbol{w} = [w_1 w_2 \dots w_7]$ denotes the weight vector to be learnt for a particular word. Also, let $\mathcal{D}$ denotes the set of words and $\mathcal{D}_s$ denotes the set of salient words present in slide S. Consider two words i and j such that $i \in \mathcal{D}_s$ and $j \in \mathcal{D} - \{\mathcal{D}_s\}$ and their visual features are $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ respectively. Then the weights learnt should satisfy the saliency ordering constraints (pairwise preferences by users): $\boldsymbol{w}^T\boldsymbol{u}_i > \boldsymbol{w}^T\boldsymbol{u}_j, \forall i, j$. For each slide S, we will have $|\mathcal{D}_s| \times |\mathcal{D} - \{\mathcal{D}_s\}|$ number of constraints. Our goal is to learn saliency ranking function $r(\boldsymbol{u}) = \boldsymbol{w}^T\boldsymbol{u}$ such that the maximum number of the following pairwise constraints are satisfied:

$$\boldsymbol{w}^T\boldsymbol{u}_i > \boldsymbol{w}^T\boldsymbol{u}_j, \forall (i,j) \in (\mathcal{D}_s, \mathcal{D} - \{\mathcal{D}_s\}), \forall S \qquad (1)$$

While the above optimization problem is a NP-hard problem, it can be solved approximately by introducing negative slack variables similar to SVM classification. This leads to the following optimization problem:

$$\begin{aligned} \min\ & (\frac{1}{2}||\boldsymbol{w}^T||_2^2 + C\sum \xi_{ij}^2) \qquad (2)\\ \text{s.t.}\ & \boldsymbol{w}^T\boldsymbol{u}_i > \boldsymbol{w}^T\boldsymbol{u}_j + 1 - \xi_{ij}; \forall (i,j) \in (\mathcal{D}_s, \mathcal{D} - \{\mathcal{D}_s\}), \forall S \\ & \xi_{ij} \geq 0 \end{aligned}$$

The above formulation is very similar to the SVM classification problem but on pairwise difference vectors, where C is the trade-off between maximizing the margin and satisfying the pairwise relative saliency constraints. The primal form of above optimization problem is solved using Newton's method [10, 17]. It should be noted that the above optimization problem learns a function that explicitly enforces a desired ordering on the saliency of words provided as training data. Now for any new word with feature vector $\boldsymbol{u}$, the saliency score can be obtained by computing the dot product of $\boldsymbol{u}$ with $\boldsymbol{w}$ (i.e., $\boldsymbol{w}^T\boldsymbol{u}$). Some example frames from different videos with the detected words and their corresponding saliency scores are shown in Figure 2. Note that the words 'Torsional' and 'Waves' are part of the title of the slide in Figure 2a and are visually more salient. Hence, they have received higher scores. Similarly, in Figure 2b, the word 'Concepts' has received the highest saliency score.

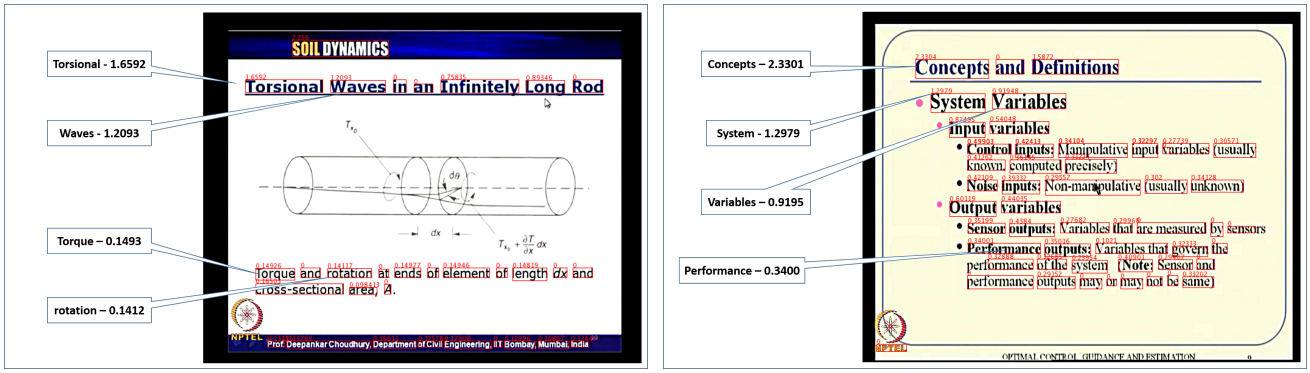
5. TOPIC TRANSITION

In this section, we discuss the steps of the topics transition part of our proposed approach. Words from different slides are clustered and unique slides are extracted before we compute probability that given slide is a topic transition slide.

5.1 Clustering of Recognized Words

The text localization and recognition in uncontrolled/wild settings is an extremely hard problem to solve. In case of educational videos, word recognition result is not always perfect and is inconsistent across the slides due to changes in lighting conditions, poor frame quality (noise and low resolution), positioning of mouse pointer over frames, presence of special symbols, punctuation, typography due to italics, spacing, underlining, shaded background and unusual typefaces. For e.g., Word 'algorithm' is recognized as 'algorlthm' in one slide and 'algorithm' in another slide. One simple approach to tackle this problem is to use a vocabulary and force the words to be one of the in-vocabulary words. However in many practical scenarios, it is often difficult to come up with a vocabulary of all words which can be present in the video (some of the technical words and proper nouns may not be present in the vocabulary). So, instead of using a vocabulary, we propose to use agglomerative hierarchical based clustering approach to cluster words that are same but recognized differently across slides.

Agglomerative hierarchical clustering [18] is a bottom-up clustering method and involves the following steps: (i) assigning each word to a different cluster, (ii) evaluation of all pair-wise distances between clusters, (iii) finding the pair of clusters with the shortest distance, (iv) merging the pair of clusters, (v) updating the distance matrix, i.e., computing the distances of this new cluster to all the other clusters, and (vi) repeat until a pair of clusters can be found with distance less than a predetermined threshold. In our system,



(a) A frame from Video1 (b) A frame from Video2

Figure 2: *Figure showing the visual saliency scores of words on few of the slides sampled from NPTEL educational videos. Note that the words which are visually more salient based on boldness, underlineness, size, location, isolation, padding and capitalization have received higher scores.*

we have used Damerau-Levenshtein distance [19] normalized by the product of the length of the two words as the distance metric (substitution, deletion and insertion cost used in the Damerau-Levenshtein distance are 1). To measure the distance between a pair of clusters, we compute the average distance (average-link hierarchical clustering) between all possible pairs of words in two clusters. Also, it must be noted that the words belonging to the same cluster will be considered as the same word for any further processing.

5.2 Unique Frame Extraction

One more novel contribution of this paper is to find out unique frames from an educational video. Unique frame extraction step finds all the unique frames (slides) in an educational video. Unique frames are identified from uniformly sampled frames of a video based on a criterion defined using pixel difference and the number of words (i.e., word clusters) matched. In case of educational videos, unique slides cannot be directly extracted by just comparing the adjacent slides as the same slide may be present in later portions of the video also (for e.g., in a typical video lecture, there will be frames of a slide followed by frames of a professor discussing the slide and then, again few frames of the same slide). Instead we compare each frame (beginning from start frame) with all the previous frames of a video and mark it as duplicate if the pixel difference threshold is less than γ or more importantly if the words overlap ratio is greater than threshold ρ with any of the previous slides. If a frame is found to be duplicate to a previous slide, it is removed from the set of possible unique slides. The pseudo code of our unique frame detection approach is provided in Algorithm 1. Using words overlap ratio along with pixel difference as the similarity metric makes our algorithm robust to change in lighting conditions, partial occlusions by the teacher and noisy video capturing methods. We note that our pipeline ignores all non-content (lecturer) frames in the video, where no text region is detected using the text detection algorithm. Hence, the output of the unique frame selection algorithm is all the unique slides present in the actual video. From this section onwards, term 'slides' or 'frames' will be used to refer to the unique slides in the video.

Algorithm 1 Finding unique frames in a video

Input: Uniformly sampled frames $\{S_m\}$, $m = 1, 2, \ldots, M$
Output: Unique frames $\{S_t\}$, $t = 1, 2, \ldots, T$ and $t \in \{1, 2, \ldots, M\}$
Approach:

```
uniqueFrames ← []
for i ← 1 ... m do
  isUnique ← true
  for j ← 1 ... i − 1 do
    if pixelDiff(S_i, S_j) ≤ γ OR wordsOverlap(S_i, S_j) ≥ ρ then
      isUnique ← false
      break
    end if
  end for
  if isUnique AND detectedWordList(S_i) ≠ ∅ then
    uniqueFrames.append(S_i)
  end if
end for
```

5.3 Content-based Features for Slides

In this subsection, we describe features which we propose to determine the topic transition probabilities. We have studied an extensive number of educational videos from different resources such as NPTEL, Coursera and EdX to figure out how a new topic is introduced in educational videos. There are two most common methods to introduce a new topic. Often a teacher while introducing a new topic, uses a few salient and novel words (the name of the new topic) in the slide. For example, the name of the new topic might be bold, placed on top of the page or might be underlined. Thus saliency of novel words definitely indicates how likely a new topic will start in a slide. Our first feature **salient word novelty** tries to capture how many novel but salient words are introduced in a slide.

Sometimes the teacher also refers to the names of the topics to be discussed later in the video by either enlisting all the topics in the video or in context with some other topics. However these occurrences usually happen with relatively lower saliency. Eventually when the topic discussion begins, the name of that topic is introduced with much higher saliency. Although these words are not novel they can still indicate topic change. Our second feature **relative saliency** is designed to capture if a word which was present earlier with lower saliency reappears in a particular

slide with higher saliency. We have found that these two features extensively cover the topic change scenarios in MOOC videos. We quantify these two features as follows:

Let us denote the unique slides obtained from previous step as set, $\mathcal{S} = \{S_1, S_2, S_3, \ldots, S_T\}$ and the words present in slide S_t as set, $\mathcal{W}_t = \{w_1^t, w_2^t, w_3^t, \ldots w_{|S_t|}^t\}$. Also, consider a function, $V : \mathcal{W} \times \mathcal{S} \rightarrow \mathbb{R}$ (where, $\mathcal{W} = \bigcup_j \{\mathcal{W}_j\}$) that takes a word and a slide as input, and returns the saliency of the corresponding word as output. For each slide, salient word novelty and relative saliency features (described below) are computed based upon the saliency of novel and non-novel words present in the slide. A word is novel with respect to a slide if it is not present in the previous few slides of a given slide, and non-novel if it is present in the previous few slides. Those previous few unique slides constitute the neighbourhood of a given slide (for e.g, if neighbourhood size is 4, then S_2, S_3, S_4, S_5 will constitute the neighbourhood of slide S_6). Let us denote the neighbourhood of slide S_t by $\mathcal{N}_t = \bigcup_{(t-|\mathcal{N}_t|) \leq j < t} \{S_j\}$ and the words present in neighbourhood as $\mathcal{W}_{\mathcal{N}_t} = \bigcup_{j \in \mathcal{N}_t} \{\mathcal{W}_j\}$. We have used $|\mathcal{N}t| = 4$ for all the videos in our experiments.

Salient Word Novelty (f_1) (for novel words): This feature is computed using only saliency of novel words present in the slide. Lets define a vector $F_t = \{V(v_1, t), V(v_2, t), V(v_3, t), \ldots\}$ such that $v_j \in \overline{\mathcal{W}}_{\mathcal{N}_t} \cap \mathcal{W}_t$ and $V(v_j, t) \geq V(v_{j+1}, t)\}$, i.e, F_t is the ordered list of only novel words sorted by their saliency scores. Then the feature f_1^t corresponding to slide S_t is computed as follows:

$$f_1^t = \mathbf{z} F_t \tag{3}$$

where $\mathbf{z}$ is weight vector. We wanted to take the number of novel words as well as their visual saliency both into account while designing this feature. We noted that the initial few (2-4) words' saliency matter most in determining new topics. If the number of novel words is high, we want our feature to ignore the saliency of all words except the first few high saliency novel words. Thus, we have used $\mathbf{z}$ as an exponential decay function which makes it more generalizable than just taking the average or maximum or sum of novel word saliency scores.

Relative Saliency (f_2) (for non-novel words): This feature is computed using relative saliency of non-novel words present in the slide. Lets define a set $\mathcal{F}_t = \{v \mid v \in \mathcal{W}_{\mathcal{N}_t} \cap \mathcal{W}_t\}$ containing non-novel words for slide S_t, then the feature f_2^t is computed as follows:

$$f_2^t = \sum_{v \in \mathcal{F}_t} \frac{\max\{V(v, j) \mid j \in \mathcal{N}_t\}}{V(v, t)} \tag{4}$$

where $\max\{V(v, j) \mid j \in \mathcal{N}_t\}$ denotes the maximum saliency of word v in neighbourhood $\mathcal{N}_t$ of slide S_t. Lower the value of this feature, higher is the chance that new topic begins here. Lower value of this feature implies that a word is present in this slide with higher saliency as compared to its neighbourhood. This feature is designed in such a way that if higher number of words reappear in a slide we reduce the topic change probability for that slide.

Bigrams. For computing features f_1 and f_2, we also use bigrams along with the individual words present in slide. A bigram is a sequence of any two adjacent words in a slide. We denote the visual saliency of a bigram as the maximum visual saliency of the two words that form the bigram. Then a bigram is treated just as another word with some saliency score, and the notion of novel and non-novel word is applicable to bigrams as well. Use of bigrams helps us in treating phrases in a systematic way.

5.4 Posterior Modelling

Once we have the 2-dimensional feature ($f^t = [f_1^t \ f_2^t]$, $1 \leq t \leq T$) extracted from each of the unique slides, posterior probability of each slide being a topic transition slide is computed. We label the topic transition slides as 1 and non topic-transition slides as 0. We use Gaussian distribution to model the likelihood. Thus, the poster distribution of a slide S_t being a topic transition slide given observation f^t is given below. First we define two Gaussian distributions which we will use to compute the posterior probability.

- $\mathcal{N}(\mu_1, \sigma_1)$: Since we want to maximize the first feature we define a Gaussian distribution centred around the maximum value of f_1^t. So $\mu_1 = \max_t(f_1^t)$ and σ_1 is set to be twice the standard deviation of f_1^t.
- $\mathcal{N}(\mu_2, \sigma_2)$: Since we want to minimize the second feature another Gaussian distribution is defined centred around the minimum value of f_2^t. So $\mu_2 = \min_t(f_1^t)$ and σ_2 is also set to be twice the standard deviation of f_2^t.

We compute the final probability as:

$$\begin{aligned} P(S_t = 1 | f^t) &= \frac{P(f^t | S_t = 1) \times P(S_t = 1)}{P(f^t)} \\ &\cong P(f^t | S_t = 1) \times P(S_t = 1) \end{aligned} \tag{5}$$

(assuming feature independence and uniform prior over slides)

$$\begin{aligned} &= P(f_1^t | S_t = 1) \times P(f_2^t | S_t = 1) \\ &= P(f_1^t | \mu_1, \sigma_1) \times P(f_2^t | \mu_2, \sigma_2) \end{aligned}$$

where $P(f_1^t | \mu_1, \sigma_1)$ denotes the probability of obtaining f_1^t from $\mathcal{N}(\mu_1, \sigma_1)$ and $P(f_2^t | \mu_2, \sigma_2)$ denotes the probability of obtaining f_2^t from $\mathcal{N}(\mu_2, \sigma_2)$. Intuitively this implies that if f_1^t is higher and f_2^t is lower for a particular slide, the posterior probability of that slide being a topic transition slide will also be higher.

6. BASELINE METHODS

In this section, we discuss the LDA based topic modelling techniques [11] that can be used for detecting topic transition points. We have used two different versions of LDA:

- **LDA:** Latent Dirichlet Allocation (LDA) is a generative model that explains the set of observations using hidden topics. In LDA, each document can be considered as a mixture of topics. In our work, each unique slide is used as a document and the visual words present in it are used as words. Each slide is assigned a topic by maximizing over the topic likelihoods obtained from LDA. Then, we find out the slides where the topic is changing from the last slide.
- **LDA with proposed saliency:** We also compare with another version of LDA where the saliency scores obtained by our approach (Section 4) are used as the weights of the words in the slides. We refer to this method as LDA with proposed saliency.

7. EXPERIMENTAL RESULTS

In this section, we evaluate our approach to detect topic transition points on publicly available NPTEL educational videos. We compare the proposed approach with well-known Latent Dirichlet Allocation based topic modelling technique [11]. We also perform a user study to evaluate the efficiency and effectiveness of our approach for finding topic starting points in educational videos and provides a quick way of navigating though videos in a non-linear fashion.

7.1 Dataset

The experiments were conducted on 10 NPTEL educational videos. The duration of each of these videos is around 1-1.5 hours; giving us total 12 hours of video content for experiments. NPTEL videos usually have a large amount of diversity. Lighting conditions, slide orientations and style, camera angle, video resolution, and lecturer positioning in the slides (for e.g., on few occasions lecturer occupies bottom right part of the slide and sometimes full frame) vary significantly across the NPTEL videos. In few of the videos, the lecturer uses printed text instead of using slides. Also, in 4 of the selected videos, along with slides, lecturer also uses handwritten text in the presentation. In 2 other videos, the lecturer writes on slides during the presentation. All these scenarios make word recognition and thus, the identification of topic transition points extremely challenging and difficult. Examples of few of the slides from different educational videos can be seen in Figure 2. Ground truth annotation of the topic transition points in this dataset are obtained from humans who are experts in the respective topics.

7.2 Evaluation

The proposed approach in this paper assigns a visual saliency score to each word in the video. The mid-level visual features extracted in Section 4.2 are combined using the weight vector obtained in Section 4.3. The weights obtained using our training set are 1.1250 (boldness), 1.0015 (location), 0.6605 (underlineness), 0.6050 (size), 0.4612 (capitalization), 0.2291 (isolation), 0.0232 (padding). We observe that boldness and location features have higher weights compared to the other feature weights indicating that these two features are perhaps more important in determining the overall visual saliency.

Next, we use these saliency scores to assign a probability for each unique slide being a topic transition slide. We generate the ranked list of slides sorted by their 'being a topic transition slide' probabilities. We compute the precision and recall for all top n elements of the ranked list, where n varies from 1 to the length of the ranked list. In our analysis, we have used F-Score to measure the performance. F-Score considers both precision and recall of the method while scoring. In this context, precision is the number of correct topic transition points retrieved (within the top n elements of the ranked list) divided by the total number of retrieved topic transition points, and recall is the number of correct topic transition points retrieved divided by the total number of ground truth topic transition topics. The F-score is defined as the harmonic mean of precision and recall:

$$\text{F-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

While the recall measures how well the system can retrieve the true ground truth topic transitions, and high precision ensures that it does not over-predict the true topic transitions, the F-Score measures the overall performance of the approach. F-Score is 1 in the ideal case (when the algorithm is perfect and when both precision and recall are 1). Following the norm regarding F-score usage[7], we also report the best F-score obtained from the ranked list. Similarly, for LDA and LDA with proposed saliency, we compute the precision and recall of the topic transitions with respect to ground truth topic transition points and get the F-Score.

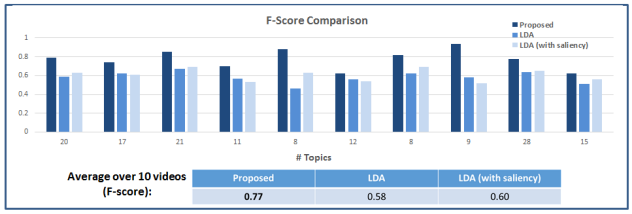


Figure 3: *Comparison of proposed approach with LDA and LDA (with visual saliency) topic modelling techniques over 10 NPTEL videos. The proposed method significantly outperforms LDA by 17%.*

In Figure 3, we provide the comparison of our approach with the LDA based techniques. We find our approach gives an average F-score of 0.77 where LDA gives an F-Score of 0.58 ± 0.018 and LDA with proposed saliency gives an F-Score of 0.60 ± 0.021 over 10 videos. The standard deviation values reported show the variation in LDA performance due to different number of topics. We vary the number of topics from 3 to 8 for both versions of LDA. Our method achieves an absolute improvement of 0.17 (relative improvement 28%) over state-of-the-art topic modelling technique LDA for topic transition detection in educational videos. Statistical significance of the improvement was also estimated using t-tests ($t(10) = 4.31$, $p = 0.0003$). This clearly shows the importance of visual saliency of words present in slides and how they can be used to detect topic transitions. We distinguish slides based on the relative saliency of their words, thus the temporal progression of saliency captures the transitions more accurately. We have also observed that the combination of two features novel word saliency and relative saliency performs the best and absence of any one of them deteriorates the performance.

7.3 User Study

We conducted a 6-participant 3-video user study to evaluate effectiveness and efficiency of the proposed system and compared it with the baseline transcript+youtube style rendering based interface (similar to the EdX interface) where the text is hyperlinked with the corresponding location in the video where it is spoken.

All the 6 participants had engineering degrees, exposure to online videos and had not seen these videos. The three videos were of 60, 49 and 56 minutes each. We design the video interface where we show the markers for topic transition points in the video timeline (Figure 4). For each video, we show the top-15 topic transition points obtained using the proposed approach. Each topic transition marker in Figure 4 corresponds to the first occurrence of the corresponding topic transition slide in the video. Each participant was presented one video with the proposed interface and one other video with the baseline interface. Thus, each video + interface combination was evaluated by two differ-

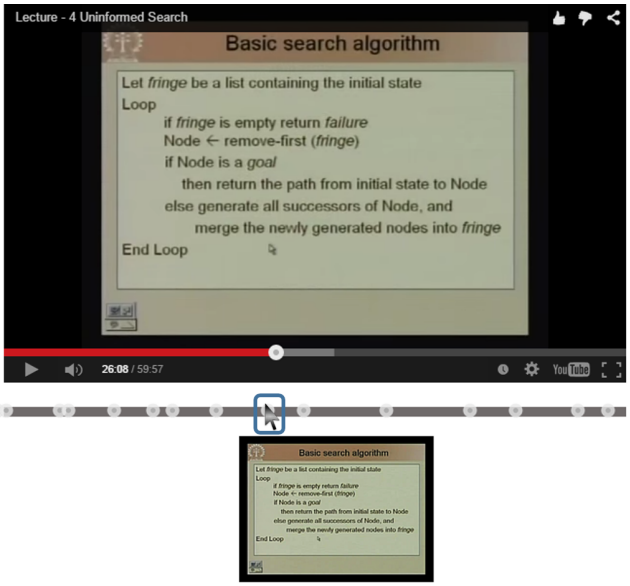


Figure 4: *Proposed video interface which shows the markers for topic transition points in the video timeline. Hovering the mouse over a marker shows the thumbnail of the corresponding topic transition slide.*

ent users. For each video, the users were given a list of 5 topics and asked to navigate to the starting point of each of these topics. They were allowed to go back and forth in the video multiple times to identify these topic locations. These 5 topics were randomly chosen from the ground truth topics given by the human experts (Section 7.1).

The total time taken by the participant to answer all the questions along with the number of correctly answered questions was measured. The answer is considered to be correct if the timestamp given by the participant is within a window of ± 10 seconds of the ground truth location. We observed that the average time taken by the participants to correctly answer one question is 50.07 ± 14.38 sec using our interface and 98.75 ± 47.75 sec using the baseline interface. The proposed interface leads to statistically significant time savings in navigating to required topics as compared to the baseline interface ($t(6) = -2.78$, $p = 0.027$). The percentage of correctly answered questions using our interface is 76.67% (out of 30 question instances) as compared to only 60% in baseline interface. Thus, the proposed interface shows both efficiency and effectiveness of our system.

8. CONCLUSION

In this paper, we propose a system for automatically detecting topic transitions in educational videos. The proposed algorithm has two novel contributions: (a) a method to assign saliency score to each word on each slide, and (b) a method to combine across-slide word saliency to estimate the posterior probability of a slide being a topic transition point. The proposed method shows a F-Score improvement of 0.17 for detecting topic transition points as compared to the LDA-based topic modelling technique. We also demonstrate the efficiency and effectiveness of the proposed method in a video navigation interface to navigate through various topics discussed in a video.

While the focus of this work is to analyze the visual content to identify topic transitions, the text transcript of the videos can also be analyzed. In the absence of manually generated text transcripts, Automatic Speech Recognition (ASR) techniques can be used. The accuracy of ASR outputs, especially given the wide variety of speaker accent and topics will be a bottleneck in their use of downstream analysis. We are currently working on combining these multiple modalities of video, speech and text to further improve the topic transition estimation.

9. REFERENCES

[1] `http://nptel.ac.in/`.

[2] C. Choudary and T. C. Liu. Summarization of visual content in instructional videos. *IEEE Transactions on Multimedia*, 9(7):1443–1455, November 2007.

[3] T. C. Liu and C. Choudary. Content extraction and summarization of instructional videos. In *ICIP*, pages 149–152, 2006.

[4] Kuldeep Yadav et al. Content-driven multi-modal techniques for non-linear video navigation. In *ACM IUI*, 2015.

[5] Dinh Q. Phung, Svetha Venkatesh, and Chitra Dorai. High level segmentation of instructional videos based on content density. In *ACM Multimedia*, 2002.

[6] Dinh Q. Phung, Svetha Venkatesh, and Chitra Dorai. Hierarchical topical segmentation in instructional films based on cinematic expressive functions. In *ACM Multimedia*, 2003.

[7] Dinh Q. Phung, Thi V. Duong, Svetha Venkatesh, and Hung Hai Bui. Topic transition detection using hierarchical hidden markov and semi-markov models. In *ACM Multimedia*. ACM, 2005.

[8] Ying Li, Youngja Park, and Chitra Dorai. Atomic topical segments detection for instructional videos. In *ACM Multimedia*. ACM, 2006.

[9] Philip J. Guo and Katharina Reinecke. Demographic differences in how students navigate through moocs. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14. ACM, 2014.

[10] Olivier Chapelle and S. Sathiya Keerthi. Efficient algorithms for ranking with SVMs. *Inf. Retr*, 13(3):201–215, 2010.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[12] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[13] Lukáš Neumann and Jiří Matas. Scene text localization and recognition with oriented stroke detection. In *ICCV 2013*. IEEE, 2013.

[14] `https://code.google.com/p/tesseract-ocr/`.

[15] `http://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html`.

[16] R. S. Wallace. A modified hough transform for lines. In *CVPR*, pages 665–667, 1985.

[17] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, pages 503–510. IEEE, 2011.

[18] A. Lukasova. Hierarchical agglomerative clustering procedure. *Pattern Recognition*, 11(5-6):365–381, 1979.

[19] `http://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein_distance`.

YouEDU: Addressing Confusion in MOOC Discussion Forums by Recommending Instructional Video Clips

Akshay Agrawal
Stanford University
akshayka@cs.stanford.edu

Jagadish Venkatraman
Stanford University
jagadish@cs.stanford.edu

Shane Leonard
Stanford University
shanel@stanford.edu

Andreas Paepcke
Stanford University
paepcke@cs.stanford.edu

ABSTRACT

In Massive Open Online Courses (MOOCs), struggling learners often seek help by posting questions in discussion forums. Unfortunately, given the large volume of discussion in MOOCs, instructors may overlook these learners' posts, detrimentally impacting the learning process and exacerbating attrition. In this paper, we present YouEDU, an instructional aid that automatically detects and addresses confusion in forum posts. Leveraging our Stanford MOOCPosts corpus, we train a set of classifiers to classify forum posts across multiple dimensions. In particular, classifiers that target sentiment, urgency, and other descriptive variables inform a single classifier that detects confusion. We then employ information retrieval techniques to map confused posts to minute-resolution clips from course videos; the ranking over these clips accounts for textual similarity between posts and closed captions. We measure the performance of our classification model in multiple educational contexts, exploring the nature of confusion within each; we also evaluate the relevancy of materials returned by our ranking algorithm. Experimental results demonstrate that YouEDU achieves both its goals, paving the way for intelligent intervention systems in MOOC discussion forums.

1. INTRODUCTION

During recent years, many universities have experimented with online delivery of their courses to the public. Hundreds of thousands of learners across the world have taken advantage of these Massive Open Online Courses (MOOCs). While MOOCs are certainly more accessible than physical classes, the virtual domain brings with it its own challenges.

Lacking physical access to teachers and peer groups, learners resort to discussion forums in order to both build a sense of belonging and to better understand the subject matter at hand. Indeed, these forums could in theory be rich reflections of learner affect and academic progress. But, with MOOC enrollments so high, forums can seem unstructured and might even inhibit, rather than promote, community [17]. It becomes intractable for instructors to effectively monitor and moderate the forums. Learners seeking to clarify concepts might not get the attention that they need, as the greater sea of discussion drowns out their posts. The lack of responsiveness in forums may push learners to drop out of courses altogether [27].

The unattended, confused learner might revisit instructional videos in order to solidify his or her understanding. Yet video, a staple of MOOCs, is tyrannically linear. No table of contents or hyperlinks are available to access material in an organized fashion. Often presented with more than one hundred ten-to-fifteen-minute videos, learners might become discouraged when they realize that they will have to re-view footage to patch holes in their knowledge.

We concerned ourselves with solving the problems related to discussion forums and videos that arise when confusion goes unaddressed. In this paper, we present YouEDU, a unified pipeline that automatically classifies forum posts across multiple dimensions, staging intelligent interventions when appropriate. In particular, for those posts in which our classifier detects confusion, our pipeline recommends a ranked list of one-minute-resolution video snippets that are likely to help address the confusion. These recommendations are computed by using subsets of post contents as queries into closed caption files. That the snippets be short is important; [10] found that, regardless of video length, learners' median engagement time with videos did not exceed six minutes. Individual learners may watch beyond the minute we recommend, should they wish.

In order to enable YouEDU's classification phase, we hired consultants to tag 30,000 posts from three categories of Stanford MOOCs: Humanities and Sciences, Medicine, and Education. The set, dubbed the Stanford MOOCPosts Dataset, is available to researchers on request [2]. Besides describing the extent of confusion, each entry in the MOOCPosts set indicates whether a particular post was a *question*, an *answer*, or an *opinion*, and gauges the post's *sentiment* and *urgency* for an instructor to respond. In detecting confusion, our classifier takes into account the predictions of five other constituent classifiers, one for each of the variables (save confusion itself) encoded in our dataset.

The online teaching platforms that Stanford uses to distribute its public courses gather tracking log data comprising hundreds of millions of learner actions. We use a subset of these data as features for our confusion classification. Some of these data are also available in anonymized form to researchers upon request [1]. Until very recently, the data requisite for our classification approach—the MOOCPosts corpus and this additional metadata—simply did not exist.

The remainder of this paper is organized as follows. We examine related work in Section 2, present the MOOCPosts corpus in Section 3, and sketch the architecture of YouEDU in Section 4. In Sections 5 and 6 we detail, evaluate, and discuss YouEDU's classification and recommendation phases. We close with a section on future work and a conclusion.

2. RELATED WORK

Stephens-Martinez, et al. [21] find that MOOC instructors highly value understanding the activity in their discussion forums. The role of instructors in discussion forums is investigated in [22], which finds that learners' experiences are not appreciably affected by the presence or absence of (sparse) instructor intervention. The study did not, however, allow for instructors to regularly provide individual feedback to learners. Instructors interviewed in [12] stress the need for better ways to navigate MOOC forums, and one instructor emphasizes in particular the benefits to be reaped by using natural language processing to reorganize forums.

Wen, et al. [24] explore the relationship between attrition and sentiment, using a sentiment lexicon derived from movie reviews. Yang, et al. [27] conduct an investigation into the relationship between attrition and confusion. While [27] also presents a classifier for confusion, our classification approach differs from theirs in that it operates on a larger dataset and uses a different set of features, including those generated by other classifiers. Chaturvedi, et al. [7] predict instructor intervention patterns in forums. Our work is subtly different in that we predict posts that coders—who carefully read every post in a set of courses—deemed to be urgent, rather than learning from posts that the instructors themselves had responded to. The classification of documents by opinion and sentiment is treated in [20] and [4].

Yang, et al. [26] propose a recommendation system that matches learners to threads of interest, while Shani, et al. [19] devise an algorithm to personalize the questions presented to learners. The need for intervention systems to address confusion in particular is highlighted in [27]. Closed caption files were used in the Informedia project [23] to index into television news shows. To the best of our knowledge, the same has not been done in the context of MOOCs.

3. THE STANFORD MOOCPOSTS CORPUS

Given that no requestable corpus of tagged MOOC discussion forum posts existed prior to our research, we set out to create our own. The outcome of our data compilation and curation was the Stanford MOOCPosts Dataset: a corpus composed of 29,604 anonymized learner forum posts from eleven Stanford University public online classes. Available on request to academic researchers, the MOOCPosts dataset was designed to enable computational inquiries into MOOC discussion forums.

Each post in the MOOCPosts dataset was scored across six dimensions—confusion, sentiment, urgency, question, answer, and opinion—and subsequently augmented with additional metadata.

3.1 Methodology: Compiling the Dataset

We organized the posts by course type into three groups: Humanities/Sciences, Medicine, and Education, with 10,000, 10,002, and 10,000 entries, respectively. Humanities/Sciences contains two economics courses, two statistics courses, a global health course, and an environmental physiology course; Medicine contains two runs of a medical statistics course, a science writing course, and an emergency medicine course; Education contains a single course, *How to Learn Math*.

Each course set was coded by three independent, paid oDesk coders. That is, three triplets of coders each worked on one set of 10,000 posts. No coder worked on more than one course set. Each coder attempted to code every post for his or her particular set. All posts with malformed or missing scores in at least one coder's spreadsheet were discarded. This elision accounts for the difference between the 29,604 posts in the final set, and the original 30,002 posts.

Coders were asked to score their posts across six dimensions:

- Question: Does this post include a question?
- Opinion: Does this post include an opinion, or is its subject matter wholly factual?
- Answer: Is this post an answer to a learner's question?
- Sentiment: What sentiment does this post convey, on a scale of 1 (extremely negative) to 7 (extremely positive)? A score of 4 indicates neutrality.
- Urgency: How urgent is it that an instructor respond to this post, on a scale of 1 (not urgent at all) to 7 (extremely urgent)? A score of 4 indicates that instructors should respond only if they have spare time.
- Confusion: To what extent does this post express confusion, or the lack thereof, on a scale of 1 (expert knowledge) to 7 (extreme confusion)? A score of 4 indicates neither knowledge nor confusion.

Coders were given examples of posts in each category. The following was an example of an extremely urgent post:

> *The website is down at the moment https://class.stanford.edu/courses/Engineering/Networking/Winter2014/courseware seems down and I'm not able to submit the Midterm. Still have the "Final Submit" button on the page, but it doesn't work. Are the servers congested? thanks anyway*

And

> *Double colons "::" expand to longest possible 0's If the longest is 0, will the address be considered valid ? (even if it doesn't make sense and there is no room for adding 0's) Can someone please answer ? Thanks in advance*

was given as an example of a post that was both confused (6.0) and urgent (5.0).

We created three gold sets from the coders' scores, one for each course type. We computed inter-rater reliability using Krippendorff's Alpha [11]. For a given post and Likert variable, the post's gold score was computed as an unweighted average of the scores assigned to it by the subset of two coders who expressed the most agreement on that particular variable. Gold scores for binary variables were chosen

	Humanities	Medicine	Education
Urgency	0.657	0.485	0.000*
Sentiment	-0.171	-0.098	-0.134
Opinion	-0.193	-0.097	-0.297
Answer	-0.257	-0.394	-0.106
Question	0.623	0.459	0.347

Table 1: Correlations with Confusion. The urgency and question variables are strongly correlated with confusion. All correlations, save the one denoted by *, were significant, with p-values < 0.01.

by majority votes across all three coders. We refer readers to our write-up in [2] for a more detailed treatment of our procedure and the complete inter-rater reliability results.

3.2 Discussion

We found significant correlations between confusion and the other five variables. In the humanities and medicine course sets, confusion and urgency were correlated with a Pearson's correlation coefficient of 0.657 and 0.485, respectively. In all three subdivisions of the dataset, confusion and the question variable were positively correlated (0.623, 0.459, and 0.347), while the sentiment, opinion, and answer variables were negatively correlated with confusion. Table 1 reports the entire set of correlations.

That questions and confusion were positively correlated supports the finding in [25] that confusion is often communicated through questions. The negative correlations can be understood intuitively. Confusion might turn into frustration and negative sentiment; as discussed in [16], confusion and frustration sometimes go hand-in-hand. If a learner is opining on something, then it seems less likely that he or she is discussing course content. And we would hope that learners providing answers are not themselves confused.

4. YOUEDU: DETECT AND RECOMMEND

YouEDU[1] is an intervention system that recommends educational video clips to learners. Figure 1 illustrates the key steps that comprise YouEDU. YouEDU takes as input a set P of forum posts, processing them in two distinct phases: (I) detection and (II) recommendation. In the first phase, we apply a classifier to each post in P, outputting a subset P_c consisting of posts in which the classifier detected confusion. The confusion classifier functions as a *combination* classifier in that it combines the predictions from classifiers trained to predict other post-related qualities (Section 5).

The second phase takes P_c as input and, for each confused post $p_m \in P_c$, outputs a ranked list of educational video snippets that address the object of confusion expressed in p_m. In particular, for a given post, the recommender produces a ranking across a number of one-minute video clips by computing a similarity metric between the post and closed caption sections. In an online system, of course, learners may choose to watch beyond the end of the one-minute snippet—the snippets effectively function as a video index.

5. PHASE I: DETECTING CONFUSION

We frame the problem of detecting confusion as a binary one. Posts with a confusion rating greater than four in the MOOCPosts dataset fall into the "confused" class, while all other posts fall into the "not confused" class. We craft a rich feature space that fully utilizes the data available in our MOOCPosts dataset, choosing logistic regression with l_2 regularization as our model.

[1]Our entire implementation is open-source.

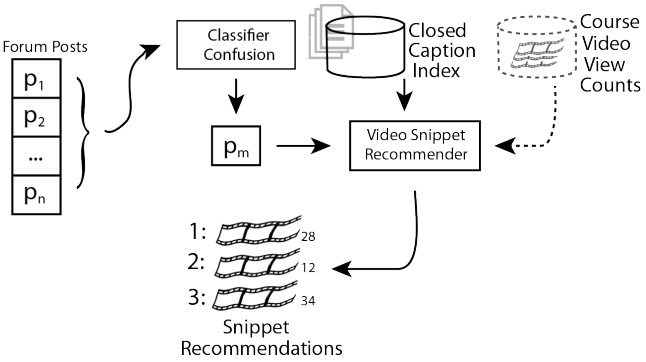


Figure 1: YouEDU Architecture. YouEDU consists of two phases: post classification and video snippet recommendation. The dotted-line module is under construction (see Section 7).

5.1 Feature Space and Model Design

Our feature space is composed of three types of inputs, those derived from the post body, post metadata, and other classifiers. The confusion classifier we train functions as a combining layer that folds in the predictions of other classifiers; these classifiers are trained to predict variables correlated with confusion. We expand upon each type of input here.

5.1.1 Bag-of-Words

We take the bag-of-words approach in representing documents, or forum posts. The unigram representation, while simple, pervades text classification and often achieves high performance [6]; we employ l_2 regularization to prevent overfitting [18]. Each document is represented in part as a vector of indicator variables, one for each word that appears in the training data. A word is a sequence of one or more alphanumeric characters or a single punctuation mark (one of {. , ; ! ?}).

Documents are pre-processed before they are mapped to vectors. We use a subset of the stop words published by the Information Retrieval Group at the University of Glasgow [14]. Words omitted from the stop word list include, but are not limited to, interrogatives, words that identify the self ("I", "my"), verbs indicating ability or the lack thereof, negative words ("never", "not"), and certain conjunctions ("yet", "but"). We ignore alphabetic case and collapse numbers, LaTeX equations, and URLs into three unique words.

5.1.2 Post Metadata

The feature vector derived from unigrams is augmented with post metadata, including:

- The number of up-votes accumulated by the post. We rationalized that learners might express interest in posts that voiced confusion that they shared.
- The number of reads garnered by the post's thread.
- Whether the poster elected to appear anonymous to his or her peers or to the entire population. It has been shown that anonymity in educational discussion forums enables learners to ask questions without fear of

judgement [9], and our dataset demonstrates a strong correlation between questions and confusion.

- The poster's grade in the class at the time of post submission, where "grade" is defined as the number of points earned by the learner (e.g., by correctly answering quiz questions) divided by the number of points possible. The lower the grade, we hypothesized, the more likely the learner might be confused.
- The post's position within its thread—we hypothesized that learners seeking help would create new threads.

5.1.3 Classifier Combination

In Section 3, we demonstrated that confusion is significantly correlated with questions, answers, urgency, sentiment and opinion. As such, in predicting confusion, we take into account the predictions of five distinct classifiers, one for each of the correlates. The outputs of these five classifiers are fed as input to a *combination function* [3]—that is, a classifier for confusion—that determines the confusion class for posts.

For a given train-test partition, let D_{train} be the training set and D_{test} be the test set. Let H_q, H_a, H_o, H_s, and H_u be classifiers for the question, answer, opinion, sentiment, and urgency variables, respectively. We call these classifiers *constituent* classifiers. Each constituent is trained on D_{train}, taking as input bag-of-words and post metadata features.

Let H_c, a binary classifier for confusion, be our combination function. Like the constituent classifiers, H_c is trained on D_{train} and takes as input bag-of-words and metadata features. Unlike the constituents, when training, H_c also treats the ground-truth labels for the question, answer, opinion, sentiment, and urgency variables as features. When testing H_c on an example $d \in D_{test}$, the constituent classifiers each output a prediction for d. These five predictions—and not the ground-truth values—are appended to the vector v of bag-of-words and metadata features derived from d. In particular, if v_h is a vector of length five encoding the predictions of the constituent classifiers, then the concatenation of v and v_h is the final feature vector for H_c.

A few subtleties: H_s uses an additional metadata feature that the other classifiers do not—the number of negative words (e.g., "not", "cannot", "never", etc.). H_q, H_a, H_u, and H_c treat the number of question marks as an additional feature, given the previously presented correlations; [27] also used question marks in predicting confusion. And while H_q, H_a, and H_o are by nature binary classifiers, H_s and H_u are multi-class. They predict values corresponding to negative (score < 4), neutral (score $= 4$), and positive (score > 4), providing H_c with somewhat granular information. Going forward, we refer to the confusion classifier that uses all the features described in this section as the *combined* classifier.

5.2 Evaluation and Discussion

In this section, we evaluate and interpret the performance of the combined classifier in contrast to confusion classifiers with pared-down feature sets, reporting insights gleaned about the nature of confusion in MOOCs along the way.

We quantify performance primarily using two metrics: F_1 and Cohen's Kappa. We favor the Kappa over accuracy because the former accounts for chance agreement [8]. Unless stated otherwise, reported metrics represent an average over 10 folds of stratified cross-validation.

Table 2 presents the performance of the combined classifier on the humanities and medicine course sets. As mentioned in Section 3, both sets are somewhat heterogeneous collections of courses, with a total of nearly 10,000 posts in each set. In our dataset, not-confused posts (that is, posts with a confusion score of at most 4) outnumber confused ones—only 23% of posts exhibit confusion in the humanities course set, while 16% exhibit confusion in the medicine course set.

5.2.1 The Language of Confusion Across Courses

Table 3 presents the performance of the combined classifier on select courses, sorted in descending order by Kappa. Our classifier performed best on courses that traded in highly technical language. Take, for example, the following post that was tagged as confused from *Managing Emergencies*, the course on which our classifier achieved its highest performance (Kappa = 0.741):

> *At what doses is it therapuetic for such a patient because at high doses it causes vasoconstrition through alpha1 interactions, while at low doses it causes dilation of renal veins and splachinic vessels.*

The post is saturated with medical terms. A vocabulary so technical and esoteric is likely only used when a learner is discussing or asking a question about a specific course topic. Indeed, inspecting our model's weights revealed that "systematic" was the 11^{th} most indicative feature for confusion (odds ratio = 1.23) and "defibrillation" was the 15^{th} (odds ratio = 1.22). Similarly, in *Statistical Learning*, "solutions" was the sixth most indicative feature (odds ratio = 1.75), and "predict" was the ninth (odds ratio = 1.65).

A glance at Table 3 suggests that our classifier's performance degrades as the discourse becomes less technical. Posts like the following were typical in *How to Learn Math*, an education course about the pedagogy of mathematics:

> *I am not sure if I agree with tracking or not. I like teaching children at all levels ... In a normal class setting the lower level learners can learn from the higher learners and vice versa. Although I do find it very hard to find a middle ground. There has to be an easier way.*

The above post was tagged as conveying confusion. The language is more subtle than that seen in the posts from *Managing Emergencies*, and it is not surprising that we saw our lowest Kappa (0.359) when classifying *How to Learn Math*. In this course, learners tended to voice more confusion about the structure of the class than the content itself—"link", "videos", and "responses" were the fourth, fifth, and seventh most indicative features, respectively.

Examining the feature weights learned from the humanities and medicine course sets provides us with a more holistic view onto the language of confusion. Domain-specific words take the backseat to words that convey the learning process. For example, in both course sets, "confused" was the

Course Set	Not Confused			Confused			Kappa
	Precision	Recall	F_1	Precision	Recall	F_1	
Humanities	0.898	0.943	0.919	0.778	0.642	0.700	0.621
Medicine	0.924	0.946	0.935	0.699	0.589	0.627	0.564

Table 2: Combined Confusion Classifier Performance, Course Sets.

Course	# Posts (% Confused)	F_1: Not Confused	F_1: Confused	Kappa
Managing Emergencies	279 (18%)	0.963	0.771	0.741
Statistical Learning	3,030 (30%)	0.909	0.767	0.677
Economics 1	1,583 (23%)	0.933	0.741	0.675
Statistics in Medicine (2013)	3,320 (21%)	0.916	0.671	0.589
Women's Health	2,141 (15%)	0.933	0.506	0.445
How to Learn Math	9,878 (6%)	0.970	0.383	0.359

Table 3: Combined Confusion Classifier Performance, Individual Courses. Our classifier performed best on courses whose discourse was characterized by technical diction, like statistics or economics. In courses like *How to Learn Math* that facilitated open-ended and somewhat roaming discussions, our model found it more difficult to implicitly define confusion.

word with the highest feature weight (odds ratios equal to 3.19 and 2.97 for humanities and medicine, respectively). In the humanities course set, "?", "couldn't", "report", "question", "haven't", and "wondering" came next, in that order. The importance of question-related features in particular is consistent with [25] and with the correlations in the MOOCPosts dataset. In medicine, the next highest ranked words were "explain", "role", "understand", "stuck", and "struggling". Table 4 displays the most informative features for the humanities and medicine course sets, as well as *How to Learn Math* and *Managing Emergencies*.

5.2.2 Training and Testing on Distinct Courses

We ran a series of experiments in which we trained the combined classifier on posts from one course and then tested it on posts from another one, without cross-validation. The results of these experiments are tabulated in Table 5.

Our highest Kappa (0.629) was achieved when training on *Statistics in Medicine 2013* and testing on *Statistics in Medicine 2014*; this makes sense, since they comprise two runs of the same course. Many instructors plan to offer the same MOOC multiple times [12]. Ideally, an instructor would tag but one of those runs, allowing an online classifier to truly shine. Yet even if such tagging were infeasible, our experience learning and testing on similar courses, such as two different statistics courses, suggests that an online classifier might well exhibit good performance. Performance might suffer, however, if the domains of the training and test data are non-overlapping, as is the case in the last two experiments in Table 5.

5.2.3 Constituent Classifiers and Post Metadata

Figure 2 illustrates the performance of each constituent classifier when cross-validating on the humanities and medicine course sets, as well as on the education course. The constituent question classifier outperformed all the others by a large margin, likely because the structure of questions is fairly consistent. Note that the constituent classifiers were not themselves fed by a lower level of classifiers; if we were attempting to predict, say, sentiment instead of confusion, we could try to improve over the performance shown here by creating a sentiment combination function that was informed by its own set of constituent classifiers.

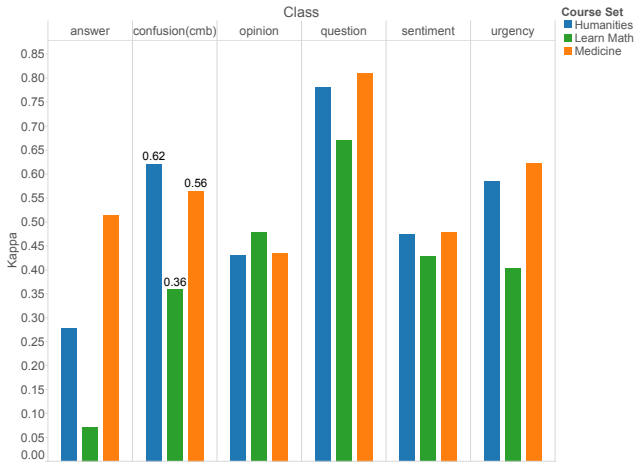


Figure 2: Constituent Classifier Performance. Confusion(cmb) is the combined classifier.

The combining function of our combined classifier consistently determined that the constituent classifiers for the question and urgency variables were particularly indicative of confusion (see Table 4). Figure 3 shows the results of an ablative analysis in which one constituent classifier was removed from the combined classifier at a time, until we were left with a classifier with no constituent classifiers (call it a *flat* classifier). The flat classifier performed worse than the combined classifier in the two course sets and the education course. For both course sets, the urgency constituent seemed to be the most helpful of the five constituents—we would expect that instructors would prioritize posts in which learners were struggling to understand the course material. However, the same was not true for *How to Learn Math*, which is consistent with the fact that no significant correlation between confusion and urgency was found (see Section 3).

The post position metadata feature also contributed positively to the classifier's performance—removing it from the flat classifier for medicine dropped the Kappa by 0.03. The other metadata features, however, did not appear to consistently or appreciably affect classifier performance, and so we chose to omit them from our ablative analysis. (Though Table 4 shows that the number of question marks was an

Humanities	Medicine	How to Learn Math	Managing Emergencies
constituent:urgency (6.59)	constituent:question (4.05)	constituent:question (6.64)	constituent:urgency (2.47)
constituent:question (3.47)	confused (2.98)	constituent:urgency (2.13)	constituent:question (2.34)
confused (3.20)	explain (2.71)	hoping (1.94)	? (1.73)
? (3.14)	role (2.41)	link (1.76)	metadata:#? (1.54)
couldn't (2.40)	understand (2.36)	available (1.63)	hope (1.40)
report (2.23)	stuck (2.27)	responses (1.62)	what (1.31)

Table 4: Most Informative Features, Odds Ratios. Features prefixed with "constituent:" correspond to constituent predictions, while those prefixed with "metadata" correspond to post metadata features. All other features are unigram words.

Training Course	Test Course	Kappa
Stats. in Med. (2013)	Stats. in Med. (2014)	0.629
Stat. Learning	Stats. 216	0.590
Economics 1	Stats. in Med. (2013)	0.267
Stats. in Med. (2013)	Women's Health	0.175

Table 5: Nature of Confusion Across Domains. Training and testing on similar courses typically resulted in high performance.

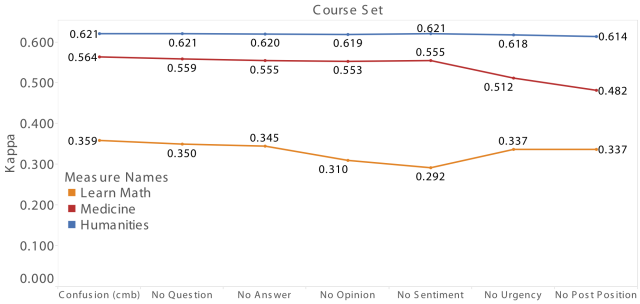


Figure 3: Ablative Analysis, Kappas. No Question is the combined classifier without the question constituent; No Answer is No Question without the answer constituent; and so on.

informative feature in the *Managing Emergencies* course.)

6. PHASE II: RECOMMENDING CLIPS

6.1 The Recommendation Algorithm

In this section, we describe how YouEDU recommends instructional material for a forum post that has been labelled as *confused* by Phase I. Every course can be thought of as a collection of several video lectures. Each video lecture on average is about 12-14 minutes long. We focus on the problem of identifying a ranked list of snippets, S, for each *confused* post. Each snippet s_i in S is a tuple *(video_id, seek_minute)* where *video_id* is an identifier for the recommended video and *seek_minute* is the time in the video to which the learner must seek and start playing the video. We would not necessarily need to recommend an *end_minute* in a deployed setting (learners could choose when to stop watching).

Phase II of YouEDU is divided into an offline indexing phase and an online retrieval phase. We define a *bin* as a time-indexed section of a video. Each bin b_i contains the transcribed text content of the video at a minute-long time interval i. We define *binscore(w, b)* of a word w and bin b as the number of times word w appears in bin b. We formulate video recommendation to learners as a classical information retrieval problem. In classical IR, the goal is to retrieve the top documents that match a user's query. In our case, the query corresponds to a confused post, and the document corresponds to a bin. We want to retrieve a ranked list of bins that addresses the content of the confused post.

6.1.1 *Offline—Indexing Pipeline*

In the indexing pipeline, we first divide each video into bins. We then use a part-of-speech tagger [5] to pre-process each bin. Nouns and noun-phrases tend to produce keywords that typically express what the content is about [13]. Hence, we represent a bin as a triplet (*video_id*, *start_min*, *noun_phrase_list)* where *noun_phrase_list* is a collection of only the nouns and noun-phrases in the bin.

We scan through each of the pre-processed bins and build an index from each word to the corresponding bin that the word appears in. This index would enable us to retrieve the list of bins B_w that corresponds to time epochs in the entire course when the word w was discussed. We also maintain a data structure that keeps track of *binscore(w,b)* for every word and bin. The constructed index and data structures are serialized to disk and are used by the retrieval phase.

6.1.2 *Online—Retrieval and Ranking:*

In the online phase, we take as input confused posts, processing each with a part-of-speech tagger. Similar to the technique we used for bins, we represent each post as a list of its constituent nouns and noun-phrases. Scanning through each of the words in the pre-processed post, we add bin b to the candidate set of retrieved bins if at least one term in the pre-processed post was mentioned in b. Since we have the index constructed offline, we can use it to prune candidates from a large number of available videos (and hence, bins) in the corpus.

We convert each post and bin into a V dimensional vector, where V is the size of the vocabulary computed over all words used in all lectures of the course. In this vector, the value on the dimension corresponding to word w_i is *binscore(w_i,bin)*. We define *simscore(P,B)* as the cosine similarity of the post and the bin.

$$simscore(P, B) = \frac{P \cdot B}{\sqrt{\sum_{i=1}^{V} P_i^2}\sqrt{\sum_{i=1}^{V} B_i^2}} \quad (1)$$

For each candidate bin C_i in the list of candidates C, we compute *simscore(C_i, post)*. We rank all bins in C by their *simscore* values and return the ranking.

6.2 Evaluation

We evaluated our ranking system on the 2013 run of the *Statistics in Medicine* MOOC, offered at Stanford University, which had 24,943 learners. We chose a random sample

of queries from our MOOCPosts dataset for that course. We ran each of those posts through Phase I of YouEDU and chose 20 random posts from the posts that were labeled as confused. For each of those confused posts our algorithm produced a list of six ranked video recommendations (that is, six bins, or one-minute snippets). We then randomized the order within each group of six, obscuring the algorithm's ranking decisions. Four domain experts in statistics at Stanford independently evaluated the relevance of each snippet to its respective post; the ratings of one expert were unfortunately lost due to technical difficulties. This process induced a human-generated ranking, which we then compared to the algorithm's rank order. The rating scale given to the raters is described below:

2: **Relevant**. The recommended snippet precisely address the learner's confusion.
1: **Somewhat relevant**. The recommended snippet is somewhat useful in addressing the learner's confusion.
0: **Not Relevant**: The recommended snippet does not address the learner's confusion.

6.2.1 Metrics

We used two metrics to evaluate the relevancy of our recommendations: NDCG and k-precision.

Normalized Discounted Cumulative Gain (NDCG): NDCG measures ranking quality as the sum of the relevance scores (gains) of each recommendation. However, the gain is discounted proportional to how far down the document is in the ranking. The underlying intuition is that the gain due to a relevant document (say, relevance score of 2) that appears as the last result should be penalized more than it would be if it appeared as the first result. Hence, the DCG metric applies a logarithmic discounting function that progressively reduces a document's gain as its position in the ranked list increases [15]. The base b of the logarithm determines how sharp the applied discount is.

If rel_i is the gain associated with the document at position i, the DCG at a position i is defined recursively as

$$DCG(i) = \begin{cases} rel_{\mathrm{i}} & i < b \\ DCG(i-1) + \frac{rel_i}{\log_b i} & otherwise \end{cases} \quad (2)$$

Since we want a smooth discounting function, we set b to 2. We use a graded relevance scale of 0, 1 and 2, corresponding to the types listed above, and computed the DCG for the ranked recommendations we obtained for each confused post. The ideal value of DCG (IDCG) is defined as the DCG based on the ideal ranking as judged by the raters. To obtain the IDCG, we sort the rankings given by the raters in decreasing order of relevance scores and compute the DCG of the sorted ranking. This corresponds to the maximum theoretically possible DCG in any ranking of the recommendations for that post. We normalize the DCG for our ranking by the IDCG to get the Normalized DCG (NDCG):

$$NDCG(i) = \frac{DCG(i)}{IDCG(i)} \quad (3)$$

If there are n recommended documents, then we report NDCG(n) as *NDCG*, the overall rating for the ranking.

Rater	NDCG	k-precision k=1	k=2	k=3
Rater1	0.66	0.66	0.61	0.62
Rater2	0.90	1.0	0.97	0.97
Rater3	0.82	0.55	0.52	0.52
Avg	0.79	0.74	0.70	0.70

Table 6: NDCG and k-Precision for recommendations

Precision at top k: We define the precision of a ranking R with n recommendations as the fraction of the recommendations that are relevant. The precision at k of a ranking R is defined as the precision of R restricted to its first k recommendations.

6.2.2 Results

Our results across the raters are summarized in Table 6. Our average precision at k=1 is 0.74. This intuitively means that on 74% of cases, the first video that we suggest to a learner (as a recommendation for his or her confused post) is a relevant video. The values at k=2 and k=3, at 0.70, are encouraging as well. Our NDCG numbers are high, indicating that we perform relatively well compared to the IDCG.

7. FUTURE WORK

The work we presented here is a first step; many opportunities for future work remain. We are actively investigating whether we can strengthen our snippet ranking further by considering which video portions learners re-visited several times. This analysis catalogs the number of views that occurred for each second of each instructional video in a course.

Another thrust of future work will use the question and answer classifiers to connect learners to each other. The challenge to meet in this work is to identify learner expertise by their answer posts, and to encourage their participation in answering questions related to their expertise. As in YouEDU, auxiliary data, such as successful homework completion, will support this line of investigation.

A third ongoing project in our group is the development of user interfaces for both instructors and learners. Using our classifiers, we have been experimenting with interactive visualizations of our classifiers' results. The hope is, for example, to have instructors see major forum-borne evidence of confusion in a single view, and to act in response through that same interface.

Video recommendations are not the only source of help for confused learners. Many online courses are repeated during multiple quarters. It should therefore be possible for our system to search forum posts of past course runs for answers to questions in current posts. Also, not all confusion is resolvable through videos. For example, difficulty in operating the video player is unlikely to have been covered in the course videos. Identifying such posts is an additional challenge.

8. CONCLUSION

We presented our two phase workflow that in its first phase identifies confusion-expressing forum posts in very large online classes. In a second phase, the workflow recommends excerpts from instructional course videos to the confused authors of these posts. Our approach utilizes new datasets of human tagged forum posts, data from learner interactions

with online learning platforms, and video closed caption files that are produced in concert with the videos for hearing-impaired learners. Evaluations of our classifiers and recommendations show that both phases of YouEDU perform well, and provide insight into the manifestations of confusion.

As novel online teaching methods are developed, the same underlying challenges will need to be met: keeping learners engaged, allowing them to feel like members of a community, and maximizing instructor effectiveness in the difficult environment of large public classes. Teaching online to very large numbers of learners from diverse backgrounds is formidable. But the potential benefits to underserved populations should encourage the investigative effort required for further research efforts.

9. ACKNOWLEDGMENTS

We sincerely thank Alex Kindl, Petr Johanes, MJ Cho, and Kesler Tannen for slogging through the snippet evaluations.

10. REFERENCES

[1] How to access the Stanford online learning data. `http://vpol.stanford.edu/research`, 2012+.

[2] A. Agrawal and A. Paepcke. The Stanford MOOCPosts Dataset. `http://datastage.stanford.edu/StanfordMoocPosts/`, December 2014.

[3] P. N. Bennett, S. T. Dumais, and E. Horvitz. The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67–100, 2005.

[4] H. H. Binali, C. Wu, and V. Potdar. A new significant area: Emotion detection in e-learning using opinion mining techniques. In *Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on*, pages 259–264. IEEE, 2009.

[5] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[6] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. Technical report, University of Washington, 2005.

[7] S. Chaturvedi, D. Goldwasser, and H. Daumé III. Predicting instructor's intervention in MOOC forums.

[8] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960.

[9] M. Freeman and A. Bamford. Student choice of anonymity for learner identity in online learning discussion forums. *International Journal on E-learning*, 3(3):45–53, 2004.

[10] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 41–50, New York, NY, USA, 2014. ACM.

[11] A. F. Hayes and K. Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

[12] F. M. Hollands and D. Tirthali. MOOCs: Expectations and reality, May 2014.

[13] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics, 2003.

[14] Information Retrieval Group at University of Glasgow. Stop word list. `http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words`. Accessed: 2015-02-05.

[15] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[16] Z. Liu, J. Ocumpaugh, and R. S. Baker. Sequences of frustration and confusion, and learning. In *Proc. Int. Conf. Ed. Data Mining*, pages 114–120, 2013.

[17] A. McGuire. Building a sense of community in MOOCs. `http://campustechnology.com/articles/2013/09/03/building-a-sense-of-community-in-moocs.aspx`, 2013. Accessed: 2015-02-01.

[18] A. Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM.

[19] G. Shani and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning.

[20] D. Song, H. Lin, and Z. Yang. Opinion mining in e-learning system. In *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*, pages 788–792. IEEE, 2007.

[21] K. Stephens-Martinez, M. A. Hearst, and A. Fox. Monitoring MOOCs: Which information sources do instructors value? In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 79–88, New York, NY, USA, 2014. ACM.

[22] J. H. Tomkin and D. Charlevoix. Do professors matter?: Using an a/b test to evaluate the impact of instructor involvement on MOOC student outcomes. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 71–78, New York, NY, USA, 2014. ACM.

[23] H. D. Wactlar, T. Kanade, M. A. Smith, and S. M. Stevens. Intelligent access to digital video: Informedia project. *Computer*, 29(5):46–52, May 1996.

[24] M. Wen, D. Yang, and C. P. Rosé. Sentiment analysis in MOOC discussion forums: What does it tell us? *Proceedings of Educational Data Mining*, 2014.

[25] N. Wilson. Learning from confusion: Questions and change in reading logs. *English Journal*, pages 62–69, 1989.

[26] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.

[27] D. Yang, M. Wen, I. Howley, R. Kraut, and C. Rosé. Exploring the effect of confusion in discussion forums of massive open online courses. In *Proceedings of the Second ACM Conference on Learning @ Scale Conference*, L@S '15, New York, NY, USA, 2015. ACM.

Seeing the Instructor in Two Video Styles: Preferences and Patterns

Suma Bhat
University of Illinois
Urbana-Champaign, USA
spbhat2@illinois.edu

Phakpoom Chinprutthiwong
University of Illinois
Urbana-Champaign, USA
chinpru2@illinois.edu

Michelle Perry
University of Illinois
Urbana-Champaign, USA
mperry@illinois.edu

ABSTRACT

Instructional content designers of online learning platforms are concerned about optimal video design guidelines that ensure course effectiveness, while keeping video production time and costs at reasonable levels. In order to address the concern, we use clickstream data from one Coursera course to analyze the engagement, motivational and navigational patterns of learners upon being presented with lecture videos incorporating the instructor video in two styles - first, where the instructor seamlessly interacts with the content and second, where the instructor appears in a window in a portion of the presentation window.

Our main empirical finding is that the video style where the instructor seamlessly interacts with the content is by far the most preferred choice of the learners in general and certificate-earners and auditors in particular. Moreover, learners who chose this video style, on average, watched a larger proportion of the lectures, engaged with the lectures for a longer duration and preferred to view the lectures in streamed mode (as opposed to downloading them), when compared to their colleagues who chose the other video style. We posit that the important difference between the two video modes was the integrated view of a 'real' instructor in close proximity to the content, that increased learner motivation, which in turn affected the watching times and the proportion of lectures watched. The results lend further credibility to the previously suggested hypothesis that positive affect arising out of improved social cues of the instructor influences learner motivation leading to their increased engagement with the course and its broader applicability to learning at scale scenarios.

1. INTRODUCTION

Lecture videos constitute the primary source of course content in the massively open online courses (MOOCs) offered by platforms such as Coursera and EdX. Not surprisingly they are also the most-used course component (compared to quiz submissions and discussion forum participation)[4, 12, 17]. Owing to the asynchronous and virtual nature of teaching and learning in these environments, lecture videos comprise the only channel through which learners have access to their instructors, an important factor affecting student motivation, satisfaction, and learning [19].

The important role of lecture videos as the primary content-bearers of a course results in instructional content designers rightly concerned about optimal video design guidelines that ensure course effectiveness; of having video lectures that maximize student learning outcomes while keeping video production time and costs at reasonable levels [9].

A recent study addresses some aspects of these concerns by comparing learner engagement patterns with video lectures across courses in the context of MOOCs [9]. The outcome of the study was a set of broad recommendations answering the concerns at a broad level. In particular, one of the take-away messages was to include the instructor's head in the presentation at opportune times by means of a picture-in-picture view of the instructor. From the perspective of this past work, our current study is a more focused version of [9]. Using the case of a Coursera course that *concurrently* made its video lectures available in two modes (the modes differ in ways in which they present a view of the instructor), the current study is unique in that it seeks to refine the recommendations made in [9]. We do this by observing how learners interact with the course in a MOOC-sized community. The central component of the current study is an empirical analysis of the course logs to highlight the differences and similarities between the motivational, navigational and engagement tendencies of the users who interact with the two available lecture modes. The uniqueness of the study is that the same set of lectures is available in two modes, which permits us to see if there are navigational behaviors and engagement patterns that are supported by specific video types.

Our empirical findings in this study are summarized below: When comparing users who watched the lectures in only one video mode,

1. We observe that learner group preferences of one mode over the other differ considerably with a ratio of 10:1.

2. Learner group preferences of the video mode for viewing lectures directly translate to differences in the pro-

portion of available lectures watched, engagement times with the videos (via differences in watch times) and in the manner in which videos are watched (streamed vs. downloaded) between the two groups.

3. Certificate earners and auditors (learners who primarily engage with a course by only watching videos) were more likely to choose one video mode over the other.

In addition, analyzing users who watched video lectures in both modes (switching twice - from one mode to the other and back to the mode first used),we notice that the disparity in preference persists (as noted above in the case of users who watched only one video mode), although the within-user differences in engagement times and the proportion of lectures watched were not statistically significant.

While many factors could be at play here, and while proposing the need for further studies to confirm our hypothesis, we posit that the video mode preferred by the majority of learners who use only one mode has the following advantage; it offers an integrated, rather than separated, access to the instructor's eye-gaze (whether the instructor is looking at the student or the content) and gestures in close proximity to the lecture content that results in a better learning experience for the learners via the availability of more realistic social cues.

2. RELATED WORK

MOOCs are criticized for their high attrition rates and are alluded to as a learning environment where a majority of students are passive lurkers who do not actively engage with the course. The low levels of engagement and completion could, in part, be attributed to the demand of the MOOC environment. MOOCs require students to be autonomous learners, who can remain motivated despite low levels of instructor presence in the course, the feeling of isolation and the unclear sense of purpose in an asynchronous learning environment. Unfortunately, aside from a handful of interactions in online discussion forums, the pre-recorded videos are the only chances for an instructor to create a sense of presence in a MOOC environment.

Prior analyses of MOOCs (e.g. [4]) have found that students spent the majority of their time watching lecture videos and that many students are auditors whose course interaction is limited primarily to watching video lectures [12]. It then follows that the design of effective videos is a critical component not only for learning effectiveness but also for the success of the course in terms of making the material accessible not just to certificate earners but also to auditors.

The design of effective video lectures, however, is informed by studies in psychology, cognitive science and online learning. Recent findings suggest that a richer instructor-student interaction in an online course is afforded by video-based sessions when compared to courses with only audio narration [3]. In addition, studies on online learning reveal that learners need to have a sense of relatedness to their instructors and that this sense is often communicated through information that is superfluous to the learning objectives [19, 5]. For instance, the presence of a humanoid pedagogical agent, be it in the form of an avatar or a cartoon figure, in a computer aided learning environment can improve a student's learning experience [6].

While the importance of non-verbal modalities of interaction (via gestures and eye-gaze) in human-human communication has long been recognized [18, 1], only recently are non-verbal modalities being harnessed in virtual communication scenarios (e.g., access to the course instructor in a window at the corner of the presentation screen in a video lecture). It is likely that increasing access to non-verbal communication can improve the instructor's sense of presence in an online-only learning environment such as a MOOC, and thus improve students' learning and their desire to stay engaged in their learning.

Clark and Mayer [6] emphasize the effectiveness of bringing instructor non-verbal modalities to the presentation because they encourage deeper engagement with the lecture content and trigger social responses in the learner [16, 7]. However, empirical evidence on its effect on learning outcomes is largely inconclusive [14, 15].

The effect of the instructor's face in visual attention, information retention and learner affect has been explored in studies such as [11, 2]. In [11] it was found that including an instructor's face in a presentation resulted in positive affective response in learners which in turn influenced the time devoted to learning. However, access to the instructor's face had no specific effect on attention or retention. In [2], an analysis of the perceptions of students being presented with two modes of video lectures incorporating the instructor's face in the presentation is available. Results suggested that having access to the instructor's gestures were potentially related to increased user satisfaction. Both these studies were not conducted in MOOC-scale environments and had a small subject pool ([11] had n=22, and [2] had n=60).

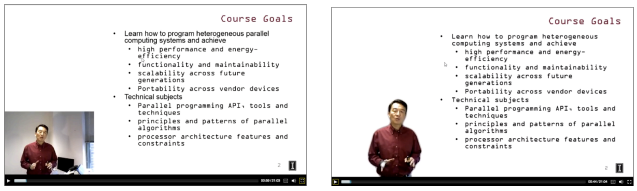
In [9] the results of a retrospective study based on course logs of MOOCs showed the effect of different video lectures produced in different styles on the engagement patterns of learners. Based on a large dataset, results indicated that video lectures that involved a talking head were more engaging to the students than lectures without a talking head. The recommendation based on these results was to include the instructor's head in the presentation at opportune times by means of a picture-in-picture view of the instructor.

This study is set with a similar goal such as that of [9] - that of understanding learners' navigational and engagement patterns with different modes of video presentations. The different modes are chosen in a way that afford access to the instructor as recommended in [9]. This permits us to see if there are navigational behaviors and engagement patterns that are supported by specific video types.

Three factors set this study apart from prior related studies. First, we compare two modes of lecture videos with access to the instructor in the *same* course. Second, the two video modes are available to the learners over a reasonable duration (three weeks/22 lectures) thus permitting the analysis over a longer duration compared to studies [11] and [2]. Third, the setting is a realistic learning at scale setting where students rely solely on video instruction.

3. METHOD

We conducted a *retrospective study* of the engagement, motivational and navigational patterns of learners as a response to video lectures presented in two styles. The learners were enrolled in the Coursera course on programming massively parallel processors offered from January to March 2014.



(a) PiP mode (b) Overlay mode

Figure 1: Screenshots of the two video modes of the lectures

3.1 Video Styles

Today's advancement in video capture technology allows for ways of improving an instructor's presence in the online classroom by including the instructor's face in the presentation at substantial reductions in video production costs. The video lectures for the course were available in two modes: the *picture-in-picture* mode and the *overlay* mode both produced in non-studio settings by the instructor and recorded simultaneously. The audio quality for both modes was excellent and similar.

Picture-in-picture mode: Presentation creation technologies can embed a video of the instructor inside a presentation, with the instructor appearing inside a window alongside the content window. In this course, the instructor window appears in the lower left corner of the presentation. We will refer to this video style as the *PiP* mode (see Figure 1a for a screenshot of this mode). The size of the instructor's video is limited by the constraints of window placement in the presentation screen.

Overlay mode: New screen capture tools are able to capture only the instructor's video without the background and overlay the video of the instructor into a presentation such as PowerPoint slides much like the green screen technology used in weather forecasts. As a result of this overlay and the screen capture technology, the instructor is able to interact with the content seamlessly by pointing at relevant sections via gestures. In addition, the instructor appears in a much closer proximity to the content window, and in a larger relative proportion compared to the instructor appearing in a window alongside the content window (PiP mode above).We will refer to this video style as the *overlay* mode (refer to Figure 1b for a screenshot). Notice how the instructor appears beside the content on the left.

The first 22 lectures, which constituted the material of the first three weeks of the course, were offered in these two modes. Both modes were available in the video lectures page on the course wiki during the entire duration of the course and were available for streamed view as well as for download. The average duration of the videos was 19.23 min. The file size of a lecture in overlay mode was about 1.2 times that of its corresponding PiP version. When the course began the course syllabus had a note about the availability of the lectures in two modes for the first three weeks and that the students were free to choose the format of their choice.

Because this was a retrospective study and not a controlled study, rather than assigning users to watch a given mode, we observed how students used the resources and interacted with them. The users[1] were classified into three groups based on the lecture modes they viewed (a user who clicked to view at least one lecture was counted in the group). There were users who viewed the lectures of the first 3 weeks only in the PiP mode (we call this group the **PiP** group, N = 899), those who viewed them only in the overlay mode (we call this group the **Overlay** group, N = 5740) and those who viewed them in both modes (the **Both** group, N = 3791). We compare the groups with respect to the analysis variables described below.

3.2 Analysis Variables

We created the following sets of analysis variables to reflect aspects of engagement, motivation and navigation.

Engagement: Because our analysis was based on the course logs, a true measurement of learner engagement is impossible. We approximate engagement via two proxy measures:

Video watching time (**wtime**): This is the total length of time that a student spends viewing video lectures (lectures 1 to 22) and we use it as the main index of engagement. This measure is limited in scope because it only provides information for streamed lecture views. Moreover, it has no indication whether the engagement with the video is an active one or a passive one (as in playing it in the background).

Discussion forum visits following a lecture view (**dfvisit**): We use a visit to the discussion forum (either to begin a thread, comment on an existing post or view a related post) immediately following a lecture (within 30 minutes) as an index of engagement. This reflects the intent of the learner to be open to aspects of the lecture beyond what is available in the video lecture.

Motivation: A limitation of this retrospective study was that access to learners' motivation (by interviewing a sample of learners, for instance) was unavailable. As a proxy to measuring motivation, we consider the following two indices:

Certificate-earner proportion (**certprop**): The fraction of users who went on to earn a certificate.

Coverage (**cov**): The fraction of lectures (and quizzes) that the learner viewed (and submitted) is our second measure of motivation. Again, an important limitation of this measure is that it only represents the fraction of lectures viewed in the streamed mode and gives no indication about those viewed after downloading[2].

Navigation: We analyzed the navigation behavior of the

[1]We only took into account users who did not explicitly drop the course.

[2]Analysis of this variable by limiting it to users who only watched a video streaming would have been a possibility but for the fact that the sample for PiP was very small (< 30).

students by observing their interaction with the course components. The measures we use are:

Streaming index (**SI**): In [12] streaming index was used as a measure of video consumption and is defined as the proportion of overall lecture consumption that occurs online on the platform (streamed), as opposed to off-line (downloaded),

$$\text{Streaming Index(SI)} = \frac{\text{streamed lecture consumption}}{\text{total lecture consumption}}.$$

Here we use it as a measure of video access.

Discussion forum activity (**dfview** and **dfpost**): The discussion forum constitutes a highly under-utilized resource in a MOOC platform and activities associated with it can be considered to be an important index of interaction with the course. Even though this measure involves a minority of course participants, we compared the number of views and posts by the users in the two groups to see if users of a video group show a tendency to participate more in discussion forums.

Back-jump proportion (**bjprop**): As used in [10], we first define a learning sequence as an ordered sequence of learning activities and its length as the number of activities in the sequence. An example of a learning sequence of length two in one session would be a lecture view followed by a quiz attempt. For our study, we consider the learning sequences of the users involving the first 22 lectures and the associated quizzes limiting the learning activities to lecture views, quiz attempts and quiz submissions.

A back-jump is a backward navigation in a learning sequence. The count of back-jumps indicates the number of times a student navigated backwards in the learning sequence and is suggestive of a departure from a linear learning sequence. In our case, this would be from a lecture to a lecture release earlier (lecture 4 to lecture 2) or from a quiz to a previous lecture (such as quiz 3 to lecture 2.3). Back-jump proportion is the number of back-jumps divided by the length of the learning sequence of the student. In [10], this measure served as an index of non-linear navigation through the course material to differentiate field-dependent learners (those who follow a sequential learning path as laid out by the content creators) from field-independent learners (those who resort to a non-linear fashion of exploring the learning environment) [8, 13], which we use in our study as well.

Other measures of comparison such as that of performance (in terms of quiz scores and assignment scores) could have been used here, but the course managed them in a server whose logs were not available in the Coursera data set.

4. EMPIRICAL OBSERVATIONS

The groups **PiP** and **Overlay** (as described in Section 3.1) are first compared with respect to the analysis variables just described and the resulting observations are summarized. Following that we analyze the users in the **Both** group.

We chose a course-week (as listed in the course wiki) as a unit and counted the number of video views during that week. In Figure 2 we see the number of unique views by the users in each of the groups during the first 3 weeks. Each

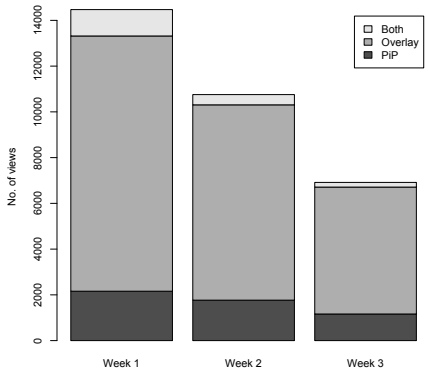


Figure 2: The number of video views in each group (Overlay, PiP and Both) over the first three weeks of the course.

bar includes the number of unique views of all lectures by a particular group during that week. What is apparent from the figure is that, over the three weeks when the lectures were available in two modes, a majority of views occurred in the Overlay mode. In addition, it is of interest to note that even in the third week there was a non-trivial number of users who watch both the modes. These views could be attributed both to the late entrants to the course and to those who switched modes in that week.

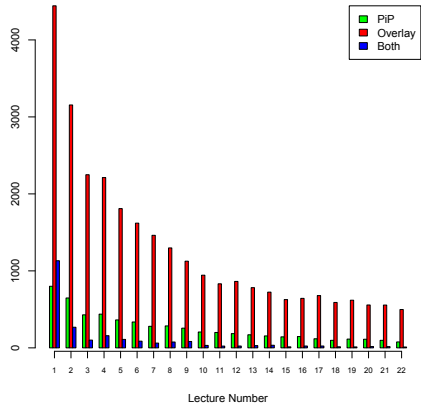


Figure 3: No. of views of each lecture over the duration of the course.

Another perspective of the views of each group is available in Figure 3 which shows the number of unique views of the 22 lectures by users in each group. Here again we notice that the *Overlay* mode was preferred by the vast majority of users compared to the *PiP* mode. It is also interesting to note from Figure 3 that the number of users who viewed the lectures in both modes is quite significant (even larger than the number of views in the *PiP* mode) for lecture 1 and then drops drastically for the lectures that follow. This could be interpreted to mean that users decide on their preferred mode as early as the first lecture. (In both these plots, the decrease in the number views is indicative of learner attrition

through the duration of the course.)

4.1 Analysis Variables Compared

We filtered out all users whose total watching time lasted less than 110s (approximating individual sessions lasting on an average shorter than 5s which could have been a result of users who paused immediately after beginning to watch a video or navigated to another page). This resulted in groups of size 385 (**PiP**), 3725 (**Overlay**) and 3791 (**Both**) respectively. Below we summarize the results upon comparing the analysis variables between the fist two groups.

A majority of the analysis variables considered here have highly skewed distributions thus deviating from the assumptions of normality. Under these circumstances, we resort to the Mann-Whitney U test to compare the two distributions. The null hypothesis tested here is not that the medians (or means) are equal but that the two groups come from the same underlying distribution. That is to say, we are testing for equality of location and shape of the distributions, not for equality of any one aspect of the distribution. Although the distributions were skewed we tabulate the mean of the variable for the two groups for the purpose of representation (see Table 1. The final column of the table indicates the p-value of the Mann-Whitney test. Statistically significant differences between groups are indicated in bold-face.

The *Overlay* and the *PiP* group: From Table 1, we observe that the underlying distributions for watch time, coverage, and streaming index differs significantly between the two groups. The **Overlay** group had a larger mean watch time compared to the **PiP** group (median watch times=33.65 min. and 21.55 min. respectively). In addition, streaming is the dominant way of accessing videos for both the groups. Streamed videos constituted an average 77% of the video usage for the **Overlay** group as opposed to 60% for the **PiP** group (respective medians 93% and 66%).

Measure	Overlay	PiP	p-value
Watch time (min)	**83.82**	**63.32**	< 0.01
Disc. forum visit	0.29	0.24	0.23
Certificate prop. (%)	8.48	6.75	0.24
Coverage	**0.24**	**0.18**	< 0.01
SI	**0.77**	**0.60**	< 0.01
Forum post	0.36	0.43	0.80
Forum view	11.86	17.22	0.59
Back-jump prop.	0.09	0.09	0.92

Table 1: Comparison of the measures for the two groups.

The 95% confidence interval of the two medians for wtime were (26.64, 38.75) for *PiP* and (49.77, 55.46) for *Overlay*. For SI the 95% confidence interval of the two medians were (0.8332, 0.8333) for *Overlay* and (0.564, 0.649) for *PiP*. Because the two confidence intervals for the medians of each group were non-overlapping, we infer that the corresponding distributions are different (also indicated by the Mann-Whitney U test).

This situation lends itself to two possible interpretations. Either more videos were watched streaming (with the same number of downloaded videos), or more Overlay videos were streamed compared to PiP with fewer Overlay videos downloaded. Both the interpretations imply that the streamed view was the primary way in which videos in Overlay mode were accessed.

As for coverage, we found that users in the *Overlay* group viewed a larger proportion of available lectures compared to their colleagues in the *PiP* group. Taken together with the lower coverage for *PiP*, its lower watch time is then justified since a smaller proportion of video views were streamed.

Although we noticed an apparent difference in the proportion of certificate earners between the two groups, a two-sample Z-test indicates that the difference in proportion was not statistically significant (p= 0.24).

Certificate Earners: We next restricted the analyses to the certificate-earners of the course, knowing that these were the most committed users in a course. The results limited to the certificate earners (N=316 for Overlay and 26 for PiP) are summarized in Table 2.

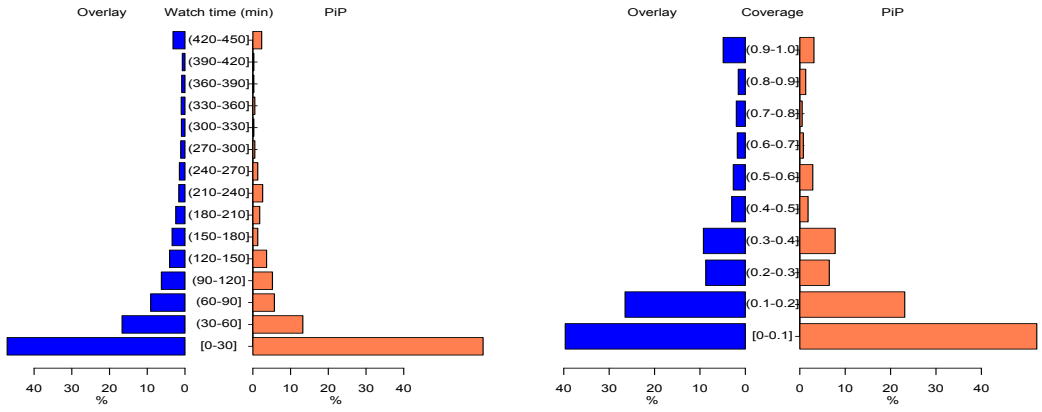
Measure	Overlay	PiP	p-value
watch time (min)	233.35	194.57	0.23
Disc. forum visit	1.53	1.69	0.84
Coverage	**0.70**	**0.58**	< 0.01
Streaming Index	**0.70**	**0.56**	0.02
Forum post	2.25	3.23	0.18
Forum view	76.44	113.08	0.08
Back-jump prop.	0.09	0.05	0.12

Table 2: Comparison of the measures for certificate earners.

We first computed the posterior probability of a certificate earner choosing one video mode over the other. Using empirical counts, we have the priors of the three groups: the probability of choosing the Overlay mode is 47%, that of choosing *PiP* is 5% and that of choosing *Both* is 48%. We also have the likelihoods: the probability that the student is a certificate-earner given that the student chose Overlay is 8.5%, the probability that the student is a certificate earner given that the student chose PiP is 6.8% (both from Table 1) and the probability that the student is a certificate-earner given that the student chose Both is 10.4% (empirically obtained).

Using this information, we calculated the probability that a certificate-earner chooses Overlay to be 0.43, that he/she chooses PiP is 0.04 and that he/she chooses Both is 0.53. This suggests that that a certificate earner is most likely to try both before settling for one mode. However, among the two modes, the more likely choice would be the Overlay mode.

Limiting the comparative analysis to the certificate earners of the two groups, from Table 2 we notice that the trends observed in the overall comparison are also largely applicable here with the exception of watch time. A surprising observation here is that despite the differences in the distributions for coverage and streaming index, differences in the distributions of the video watching times were not statistically significant. A likely explanation is that the certificate earners in the PiP group revisited portions of the same video, resulting in longer watch times compared to their Overlay



(a) Distribution of Watch times

(b) Distribution of Coverage

Figure 4: Histograms of Watch time (left) and Coverage (right) for the two groups compared. Each plot shows the density corresponding to each bin in the y-axis.

colleagues.

What is new here is that certificate earners in the *Overlay* group show apparently different non-linear navigational patterns compared to their *PiP* counterparts as evidenced by the difference in means. However, the distribution of back-jump fractions is not statistically significant (p=0.12) possibly owing to the relatively small sample size of the PiP certificate earners (n=26).

Auditors: From [12] we know that auditors (defined in that study as learners who did assessments infrequently if at all and engaged instead by watching video lectures) are nearly as engaged and motivated in the course as certificate earners in terms of using lecture materials in MOOCs and show similarly high levels of overall learning experience to certificate earners. Here, we investigate the extent to which users in the two groups had engagement levels similar to that of certificate earners.

We identified the auditors by clustering the users using k-means in the Overlay and the PiP groups by three factors into 3 classes (certificate earners, auditors, and lurkers):

- coverage (answering the question 'How many lecture units were watched?');
- streaming index (answering the question 'How were the lectures watched?');
- watch time (answering the question 'For how long were the lectures watched?').

We observed that the certificate users fell into a predominant group, which also included a set of non-certificate users 'similar' to the certificate users; these users behaved like the certificate users with respect to the 3 factors considered here. We refer to these users as auditors since they used resources much like the certificate users, except for the fact that they did not earn a certificate. We noticed that 3.5% of Overlay users were auditors in this sense and nearly 6% of users in the PiP were auditors. The difference in proportion of auditors was statistically significant (p=0.012), suggesting that PiP had a larger proportion of auditors compared to Overlay.

We then calculated the likelihood of an auditor choosing a specific viewing mode using empirical counts and note that the probability that an auditor chooses Overlay was 0.86 much greater than the probability that an auditor chose PiP, which was 0.14.

4.2 The Both group

While a comparison between the Overlay and the PiP groups served as a type of between-subjects analysis, a within-subjects type of analysis is afforded by analyzing the Both group. Although users watched both video modes in this group, to get a more reliable picture of engagement patterns and video mode choices, we included only those users who watched at least half of all the available lectures. With this set-up we assume that the users had sufficient exposure to the mode in which they began watching lectures before switching to the other mode. In addition, they had sufficient opportunities to experience the second mode and revert back to the original mode if they chose to do so.

Users in this group watched lectures in both modes and could be divided into three groups: 1) those who viewed a set of lectures in one mode and then switched to the other mode and remained in that second mode for the rest of the lectures, 2) those who switched twice eventually returning to watch the remaining lectures in the original mode in which they began, and 3) those who showed no apparent preference for one mode over another. For the purpose of our analysis, we focus on the second of these three groups because the sample size of the first group was too small (< 30) to draw meaningful inferences and we had no meaningful analyses to conduct with the third group.

With this restriction on the users, we were left with 271 users (34% of the users in Both), of which 241 (89%) watched

	OPO	POP	p-value
Coverage	0.71	0.61	< 0.01
Streaming Index	0.80	0.57	< 0.01
Watch time (min)	291.69	260.85	0.10
Disc. forum visit	1.83	1.61	0.56
Back-jump prop.	5.6	4.6	0.15
Certificate prop.	0.37	0.63	<0.01

Table 3: Comparison of the mean values of the measures for the users in the *Both* group.

most of the lectures in the overlay mode and the remaining 30 watch most of the lectures in the PiP mode. It is clear that the majority of users in this group began watching the lectures in the overlay mode, switched to the PiP mode, and reverted to watching in the overlay mode. We represent this majority group as OPO and the other group as POP. For each user in the POP and OPO groups, we computed the measures of coverage, streaming index and watching time over the lectures watched in a given mode, yielding a measure for each video mode watched. We summarize these measures in Table 3.

We observe from Table 3 that the distributions of coverage and streaming index for the Overlay mode and PiP mode differ substantially and that the difference is statistically significant. We infer that a larger proportion of lectures were watched by the users following an OPO pattern compared to a POP pattern and that the videos in Overlay mode were streamed, while the videos in PiP mode were mostly downloaded. We we notice that the distributions of watch times were not different between the OPO and POP. This implies that when the users had a chance to watch both the modes, their engagement patterns with their 'preferred' mode was similar.

Unlike in the case of the groups that watched only one mode, a comparison of the proportion of certificate earners between the two Both groups shows that a larger proportion of POP were certificate earners and that the difference in proportion was statistically significant via a two-sample Z-test ($p < 0.01$).

5. INTERPRETATION OF RESULTS

The present study suggests that learners showed a strong preference for the Overlay mode over the PiP mode. Comparing the user groups that viewed the lectures in only one mode, we saw that the two groups differed significantly in their watching times, choice of video access and proportion of lecture materials viewed. The preference of Overlay was also exhibited by the users that watched both modes. This suggests that the Overlay mode was preferred and we hypothesize that these videos appeared more engaging. Taken in light of the results of studies such as [7], the findings here could be interpreted to mean that this was the result of a positive affective response of the learners to social cues in the learning environment (here the videos). It is likely that the overlay mode offered several affordances over the PiP mode – integrated rather than separated access to the instructor's eye-gaze and gestures, the instructor's proximity to the slides, and the larger size of the instructor – which could have yielded differences in social cues available via the video modes.

This primary social cue that was different between the two video modes, we hypothesize, was the integrated view of a real instructor and this is likely to have increased learner motivation, which then affected the amount of time learners spent watching a lecture and the proportion of lectures they watched. Aside from this hypothesis on the difference in the availability of social cues, in the absence of watching actual behaviors of the learners affording a more fine-grained characterization of their watching patterns (such as the actual time users spent watching the video or the amount of time they spent looking at the instructor's face) and a qualitative analysis via interviewing users for their opinions about the videos, the true implications of the difference on the video watching/consuming patterns cannot be determined. Another set of experiments to quantify the differences more specifically in terms of the perceptions of the students via qualitative and quantitative measures is currently underway and the results will be a valuable extension to the results of this study.

Based on empirical estimates of likelihood and priors, both certificate earners and auditors, two groups most engaged with the lectures, showed a higher chance of choosing the Overlay mode suggesting the possibility of this mode being conducive to the viewing characteristics of these learners. The higher chance of a certificate earner choosing the overlay mode over the PiP could be interpreted to mean that improved access to instructor's presence is important to even the most motivated of users of a course in a MOOC environment.

6. LIMITATIONS AND FUTURE WORK

A primary limitation of this study is the lack of a qualitative analysis of user affect and satisfaction with the video mode of their choice. In the absence of the qualitative dimension to our study, most of the quantitative analysis were done based on proxy measures of motivation and navigational intent. Moreover, the measures chosen for the quantitative comparison were approximations based on the course logs with their inherent limitations. A more controlled study encompassing both qualitative aspects and more representative measures of engagement and navigation would shed more light on design guidelines for video lectures.

Our primary measure of engagement, video watching time, only measured the overall interaction with videos without regard to the finer engagement patterns such as the number of pauses and restarts, segments revisited, and playback rate changes that characterize a video view session. Incorporating these details as part of engagement patterns will offer a more refined view of patterns of engagement that are supported by different video presentation styles.

Other aspects for future work in this context would be exploring the preferences based on differences in demographic backgrounds of learners[3]. This would offer key insights about the preferences of a global audience that MOOCs aspire to

[3]Although learner IP address information was available, their potential of being considered as personally identifiable information precluded their inclusion in the analyses.

serve. Another important direction for future work is to explore if the same preferences and outcomes would arise regardless of the demographics the course topic attracts and the immediate functionality of seeing the instructor clearly (i.e content/topic specificity of the course).

7. CONCLUSION

Recognizing the important role that lecture videos play as primary content-bearers of a course in MOOCs, instructional designers are justified in their concerns about the kinds of video presentations that lead to best learning outcomes, keeping video production costs at reasonable levels. In this study we compared two video modes that offered the same set of lectures for a significant duration of a course in programming parallel processors. We found that a significantly large proportion of learners preferred one mode over the other. We hypothesize that the modes primarily differed in their ability to make the instructor's gaze and gestures more directly accessible to learners and that the mode that offered more access to instructor's gestures and eye-gaze was probably the preferred mode by the vast majority of learners. We also hypothesize that these users, possibly owing to the resulting positive affect created by improving the instructor's social presence, showed more engagement with the videos (via larger watch times), preferred the streamed mode of viewing videos (indicating immediacy in user response) and covered a larger proportion of lectures. The results also support the possibility that certificate earners (the most motivated of learners) and auditors (learners who primarily engage with a course by only watching videos) showed a higher chance of choosing the video mode offering better access to instructor's gaze and gestures, suggesting that the mode is perhaps conducive to the viewing characteristics of these learners.

8. REFERENCES

[1] M. Argyle. *Bodily communication*. Routledge, 2013.

[2] S. Bhat and G. Herman. Student perceptions of differences in visual communication mode for an online course in engineering. In *Frontiers in Education Conference, 2013 IEEE*, pages 1471–1473. IEEE, 2013.

[3] J. Borup, R. E. West, and C. R. Graham. Improving online social presence through asynchronous video. *The Internet and Higher Education*, 15(3):195–203, 2012.

[4] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. Seaton. Studying learning in the worldwide classroom: Research into edx?s first mooc. *Research & Practice in Assessment*, 8:13–25, 2013.

[5] M. C. Carlisle. Using you tube to enhance student class preparation in an introductory java course. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 470–474. ACM, 2010.

[6] R. C. Clark and R. E. Mayer. *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons, 2011.

[7] G. Cui, B. Lockee, and C. Meng. Building modern online social presence: A review of social presence theory and its instructional design implications for future trends. *Education and information technologies*, 18(4):661–685, 2013.

[8] N. Ford and S. Y. Chen. Matching/mismatching revisited: an empirical study of learning and teaching styles. *British Journal of Educational Technology*, 32(1):5–22, 2001.

[9] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.

[10] P. J. Guo and K. Reinecke. Demographic differences in how students navigate through moocs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 21–30. ACM, 2014.

[11] R. F. Kizilcec, K. Papadopoulos, and L. Sritanyaratana. Showing face in video instruction: effects on information retention, visual attention, and affect. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2095–2102. ACM, 2014.

[12] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179. ACM, 2013.

[13] J. O. Liegle and T. N. Janicki. The effect of learning styles on the navigation needs of web-based learners. *Computers in Human Behavior*, 22(5):885–898, 2006.

[14] R. E. Mayer. 14 principles based on social cues in multimedia learning: Personalization, voice, image, and embodiment principles. *The Cambridge Handbook of Multimedia Learning*, page 345, 2014.

[15] R. E. Mayer and C. S. DaPra. An embodiment effect in computer-based learning with animated pedagogical agents. *Journal of Experimental Psychology: Applied*, 18(3):239, 2012.

[16] B. Reeves and C. Nass. The media equation: How people respondto computers, television, and new media like real people and places. 2010.

[17] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? *Communications of the ACM*, 57(4):58–65, 2014.

[18] R. Vertegaal, G. van der Veer, and H. Vons. Effects of gaze on multiparty mediated communication. In *Graphics Interface*, pages 95–102, 2000.

[19] A. Wise, J. Chang, T. Duffy, and R. d. Valle. The effects of teacher social presence on student satisfaction, engagement, and learning. *Journal of Educational Computing Research*, 31(3):247 – 271, 2004.

Using Partial Credit and Response History to Model User Knowledge

Eric G. Van Inwegen Seth A. Adjei Yan Wang Neil T. Heffernan
100 Institute Rd
Worcester, MA, 01609-2280
+1-508-831-5569
{egvaninwegen, saadjei, ywang14, nth} @wpi.edu

ABSTRACT
User modelling algorithms such as Performance Factors Analysis and Knowledge Tracing seek to determine a student's knowledge state by analyzing (among other features) right and wrong answers. Anyone who has ever graded an assignment by hand knows that some answers are "more wrong" than others; i.e. they display less of an understanding of the skill(s) involved. This investigation seeks to understand the effects of progression through wrong answers to right answers in a way to determine how the "level" of wrongness affects future performance. The key findings are that A.) where in a series of opportunities a student reaches the goal impacts future performance, as does B.) the "level" of previous wrongness, even two questions before the current opportunity.

Right students are all alike;
every wrong student is wrong in his or her own way.
(with apologies to Ms. Karenina and Mr. Tolstoy)

1. INTRODUCTION
The use of algorithms to estimate student knowledge based on performance on intelligent tutoring systems (ITS) has been around for two decades. Two of the more well-known methods are knowledge tracing (KT) [6] and performance factors analysis (PFA) [11]. Both models use a student's right or wrong answers and develop a model to estimate the chance that a student has "learned" a particular skill. KT uses Bayes nets to determine four parameters per skill; PFA uses logistic regression to determine three parameters per skill. Although the order of correctness is incorporated into the models, both use only correctness as their input. Other pieces of information that may be collected by the ITS are neglected in these models.

ITS may collect any number of additional pieces of information about a student, their actions, their exact answers, etc. For example, Baker et. al. use over 20 features to make their predictions [2]. Some even make use of biometrics through additional sensors. (See Cavalio and D'Mello's review of several methods [3]. The goal of many of these algorithms is to try to make a computer tutor that is at least as responsive, observant, and effective as a human tutor would be. Incorporating more data about a student's affect can be seen as an attempt to give a computer access to the information that a human tutor would notice. However, the more detailed that a model becomes, the more computationally time-consuming it becomes. Also, as the number of inputs increases, fewer ITS's can make use of it (as a complex set of inputs may not be collected on all systems). One feature that might be incorporated into these algorithms is the use of the number of attempts and hints a student uses to answer a problem to classify more conditions than binary right and wrong and to look for the effect of how long it takes a student to achieve a particular classification.

Human teachers often employ the idea of partial credit, both as a motivational tool, and as a more accurate measure of knowledge (when compared to the binary correctness). Any teacher who has graded papers knows that some wrong answers (and workflow) demonstrate a nearly full understanding of a skill, while other wrong answers demonstrate a near-total lack of understanding. The idea of using dynamic testing (that is, a testing medium that gives hints to and tracks the number of attempts made by students) has been around since at least the 1980's. Bryant, Brown and Campione [5] compared traditional testing (binary correctness) to dynamic testing (tracking how many hints students needed to be successful). Others (e.g. Grigorenko and Sternberg) reviewed this kind of dynamic testing (among other methods) [8] and concluded that dynamic testing provides a more accurate measure [12]

Unfortunately, some ITS's can only determine the "worthiness" of a wrong answer if all wrong answers are somehow programmed in. Some ITS's do make use of pre-programmed wrong answers, but partial credit may or may not be given. Efforts before this one have been made to use partial credit to measure student knowledge [16]. E.g., in ASSISTments[1], wrong answers may be programmed to give a student a particular message, but A.) students are still marked completely wrong (and given no credit) and B.) all of these wrong answer messages must be programmed into the problems (which is incredibly time-intensive).

A more common method of assigning partial credit in ITS's is to give partial credit based on the number of attempts it takes a student to get the right answer [1] and/or the number of hints a student uses [9]. This is much faster to program, and does not require looking at all possible wrong answer to determine which ones show a limited understanding of the skill (as opposed to no understanding of the skill). The basic argument would be that a student who is "only slightly wrong" might figure out her mistake

[1] ASSISTments is an online learning system primarily for math, based out of Worcester Polytechnic Institute.

after only one wrong attempt, while a student who is "very wrong" might need several hints and several attempts before he can get the problem right. We are not analyzing specific wrong answers in this treatment; we using a student's partial credit history to modify the probability of that student getting the next question correct.

In this paper, we are analyzing a dataset from ASSISTments from the years 2012-2013. (The dataset contains ~ 500K student-problem instances; the content is mainly middle-school mathematics.) We analyze the student entries for patterns of attempts, hints use, and a simplistic order of actions to determine "bins" of students. We are also able to analyze the data to seek patterns of moving through bins (that is, as a single student uses more or less assistance on subsequent problems), and when in a particular opportunity count a bin (or sequence of bins) is encountered. We build off of our earlier work presented at the Learning Analytics & Knowledge Conference, 2015.

1.1 Background

In our previous work [13], we built off of other works that looked at attempt use, hint use (Assistance Model – AM – [15]), and simple sequence of action (Sequence of Action model – SOA – [7 and 17]), and modified and combined these models to make our own. We looked at the combination of number of attempts used to get the right answer, hint use, whether the "bottom-out hint" (BOH) was used, and a simplistic order of actions. In our model, the values for each parameter were:

- attempt use: 1, 2, 3, 4, (5+)
- hint use: 0, 1, 2, (3+)
- first action: hint or attempt
- BOH: used or not used

This gave us 35 different combinations. By analyzing the similarities of actions and future performance - defined as the average next problem correctness (NPC) and found by using pivot tables on 80% of the dataset, the 35 bins were combined into only 16. This gave us the "Fine-Grain Action" model (FGA). Table 1 shows the bins and re-grouped bins, and the NPC values.

Table 1a: The Fine-Grain-Action model 1st action = attempt

<table>
<tr><th></th><th>1 att.</th><th>2 att.</th><th>3 att.</th><th>4 att.</th><th>5 + att.</th></tr>
<tr><td>0 hint</td><td>0.8156
Bin 1</td><td>0.7380
Bin 2</td><td>0.6771
Bin 3</td><td>0.6380
Bin 4</td><td>0.6211
Bin 5</td></tr>
<tr><td>1 hint</td><td>-----</td><td colspan="2">0.7012
Group A</td><td colspan="2">0.6321
Group C</td></tr>
<tr><td>2 hint</td><td>-----</td><td colspan="4" rowspan="2">0.5812
Group E</td></tr>
<tr><td>3+ hint</td><td>-----</td></tr>
<tr><td>BOH</td><td colspan="5">0.5099
Group G</td></tr>
</table>

Table 1b: The Fine-Grain-Action model 1st action = hint

<table>
<tr><th></th><th>1 att.</th><th>2 att.</th><th>3 att.</th><th>4 att.</th><th>5 + att.</th></tr>
<tr><td>0 hint</td><td>-----</td><td>-----</td><td>-----</td><td>-----</td><td>-----</td></tr>
<tr><td>1 hint</td><td>0.7083
Bin 6</td><td colspan="2">0.6192
Group B</td><td colspan="2">0.5702
Group D</td></tr>
<tr><td>2 hint</td><td>0.5250
Bin 11</td><td colspan="4" rowspan="2">0.4688
Group F</td></tr>
<tr><td>3+ hint</td><td>0.4118
Bin 16</td></tr>
<tr><td>BOH</td><td colspan="5">0.3396
Group H</td></tr>
</table>

1.2 Research Questions

Extending from our previous analysis, we have three questions we want to address here:

1.) What is the significance of the bins?
   a) What is the statistical significance of the different bins? E.g. are bins "x" and "y" (arbitrary names) reliably different?
   b) Can the bins be re-grouped into larger groups without loss of predictive power? (E.g. Why 16? Why not 35 or 3?)
2.) Can the sequence of students moving through "Super Bins" be used to make more accurate predictions? (E.g. Is there a difference in expected outcome when comparing a student who moves from Super Bin 3 to 1 vs. 5 to 1?)
3.) Should all wrong answers be treated equally? Can we use reasonably simple and replicable methods to identify what student actions demonstrate different levels of understanding of the material?
   a) Is there an impact of bin sequence and / or opportunity count on predicted outcome?

2. METHODS

2.1 Creating the "SuperBins" (Method 1)

A quick glance at the next problem correctness (NPC) values in Table 1 shows that some bins are very nearly equivalent. When displayed in the above format, local values vary enough to warrant the bins. However, when put in order by bin values (which are just the mean NPC for instances falling into that category), we can now run a simple t-test (two tailed) analysis to compare one bin to the one that comes immediately after. This gives us Table 2.

Table 2: The bins from the FGA reordered and showing the p-value that compares one bin to the one immediately below.

Bin	NPC	stdev	n	p-value	Ordinal
1	0.8156	0.3878	215,870	< 0.0001	1st
2	0.7380	0.4397	22,229	0.0055	2nd
6	0.7083	0.4545	1,958	0.5827	3rd
A	0.7012	0.4577	3,414	0.0162	4th
3	0.6771	0.4676	5,616	0.0009	5th
4	0.6380	0.4806	2,326	0.7168	6th
C	0.6321	0.4822	1,408	0.4941	7th
5	0.6211	0.4851	2,518	0.9416	8th
B	0.6192	0.4856	407	0.1339	9th
E	0.5812	0.4934	4,011	0.8154	10th
D	0.5702	0.4950	114	0.3782	11th
11	0.5250	0.4994	541	0.4851	12th
G	0.5099	0.4999	40,652	0.0781	13th
F	0.4688	0.4990	465	0.1252	14th
16	0.4118	0.4922	289	0.0141	15th
H	0.3396	0.4736	13,989	-----	16th

In Table 2, the p-value analysis comparing the bin of that line to the one below it allows us to identify natural break points and groups. Bins are regrouped according to these break points. That

is, bins are grouped together as long as two bins fail to be statistically different. This gives us five "SuperBins" (Table 3).

It may seem somewhat arbitrary to keep bins 16 and H separate (with a p-value of 0.0141), while grouping A and 3 together (with a p-value of 0.0162). We could argue that we used a deciding value of 0.015, but that would be an arbitrary value. The real reason for keeping 16 and H separate is that the action of using the bottom out hint (and using a hint as the first action) seems to be different than any other combination of actions and should be kept separate. Throughout the rest of this analysis, we will see that the results of keeping this bin separate as its own SuperBin gives us more predictive ability.

This gives us a useful and relevant way to regroup bins that are not reliably different. One can easily make the argument against the 16 bins in FGA that, if two bins are not statistically different, why have them? By combining statistically similar bins, there is more meaning (in prediction) to assigning a particular value for the next problem correctness, even if the recombination "smooths over" the different ways that a student could arrive at a particular prediction.

Table 3: The five "SuperBins" with their predictive values, and relevant statistics. The colors are used consistently throughout the paper for clarity sake.

SuperBin	NPC	stdev	n	p-value
1	0.8156	0.3878	215,870	<< 0.0001
2	0.7380	0.4398	22,229	<< 0.0001
3	0.6902	0.4624	11,015	<< 0.0001
4	0.5297	0.4991	52,731	<< 0.0001
5	0.3396	0.4736	13,989	----

If we use the colors to remake a condensed Table 1, we can see that the SuperBins are locally consistent within the FGA. This is significant in that it suggests that, although many of the 16 bins from FGA may be statistically similar, these similarities (and differences) occur logically throughout the chart. (See Table 4.)

Table 4: FGA color coded according to SuperBins.

Hints	1 att.	2 att.	3 att.	4 att.	5+ att.
0	Bin 1, 0.816	Bin 2, 0.738	Bin 3, 0.677	Bin 4, 0.638	Bin 5, 0.621
1	Bin 6, 0.708	Grp A 0.701		Grp C 0.632	
		Grp B 0.619		Grp D 0.570	
2	Bin 11 0.525	Grp E 0.581			
		Grp F 0.469			
3+	Bin 16 0.412	Grp E			
		Grp F			
BOH	**attempt 1st**	**Grp G 0.510**			
	hint 1st	**Grp H, 0.340**			

It is also worth noting that, although the bin numbers that went into the SuperBins may seem random, there is a pattern. SuperBin 1 consists of students who get a problem right. SB2 is populated by only students who made only one wrong attempt (and used no hints) before getting the answer right on their own. SB3 comes from three bins that represent only a small number of attempts / hint use. SB4, which incorporates the bulk of the FGA bins, is anything left, except for using the bottom-out hint, with the first action being hint use. We can now use these SuperBins as the identifier of "wrongness".

Table 5: Meaning (in terms of attempt and hint use) and interpretation of "wrongness" of the five SuperBins

SuperBin	Meaning	"Wrongness"
1	Student got it right	Right
2	Student made one wrong attempt, and then got it right.	Barely wrong
3	Student used a few attempts, and 0 or 1 hint.	Partially wrong
4	Student used many attempts and/or hints.	Significantly wrong
5	Student could not start without a hint, and needed the answer.	Completely wrong

In ASSISTments, a *must* get the right answer before moving onto the next question, no matter how many attempts they make or hints they use. Clearly, a student who makes one wrong attempt and then gets the answer right with no hints demonstrates that their thinking was "less wrong" than a student who makes a series of attempts and uses many hints before getting to the correct answer. SuperBins give us a working definition of "wrongness".

2.2 Impact of previous bin; 2 SuperBin (2SB) combinations (Method 2)

Looking at the sequence of students "moving" through SuperBins can help us to better understand how a student's knowledge on a skill is changing. As we look at a student's performance on one skill, progression through SuperBins would indicate that the student's knowledge is improving; most humans would call this "learning." Likewise, a student who gets an answer right, and then regresses could have "slipped" (to use KT terminology loosely).

The first (and simplest) method to look at the impact of previous SuperBins on future success is to look at two-bin combinations. That is, after the first problem, we will look at not just the SuperBin a student falls into on opportunity n, but also the SuperBin they were in on opportunity (n-1). This gives 25 different combinations. Our naming convention is (current).(previous). Thus, 2.1 is a student who is in SuperBin 2 (used one wrong attempt before getting a problem right on the second try) and was in SuperBin 1 (got the problem right on the first attempt). To use knowledge tracing language, 2.1 could represent a "slip". Two-SuperBin code 1.2 is a student who was in SuperBin 2 and has improved to SuperBin 1. Two-SuperBin codes run from [(1.1-1.5) - (5.1-5.5)].

Table 6 (next page) illustrates the impact of the previous question's "wrongness" on the outcome after the current question. For instance, if we compare the values of the 1.x family, we should not be surprised that the 1.1 (two correct in a row) has the highest probability of success on the next problem. However, the four other two-bin combinations (1.2-1.5) all have (statistically significantly) different predictions for the next problem. That is, how wrong a student was on the previous question can be an indicator for how likely they are to get a question right, even after they have gotten one right.

Perhaps the best demonstration of the importance of using a partial credit metric (of some sort) is to compare the predicted outcomes for 2.2 and 5.5. In both cases, the students would be marked wrong on two consecutive problems. However, a student who manages to make a mistake and then correct themselves with no

aid (twice) is (un-surprisingly) much more likely to get the next problem correct than one who needs the answer given to them (and won't even start without a hint). A student in 2.2 has a nearly 70% chance of success on the next problem, while a student in 5.5 has a mere 16.7% chance! Without looking at partial credit, they would be marked equally wrong.

Table 6: Two SuperBin Combinations. Code 1.x refers to students who are currently in SuperBin 1 and who were in SuperBin x on the last problem. "Families" (1.x, 2.x, etc.) are color coded according to current SuperBin. Codes without decimal (bolded) are values from Table 3. The p-values compare a 2SB to the one below it.

2SB	NPC	n	p-value
1	**0.816**	**215,870**	
1.1	0.840	121,317	< 0.0001
1.2	0.806	15,440	< 0.0001
1.3	0.775	7,085	< 0.0001
1.4	0.703	26,109	< 0.0001
1.5	0.655	4,317	-----
2	**0.738**	**22,229**	
2.1	0.783	11,421	< 0.0001
2.2	0.699	2,137	0.5322
2.3	0.688	1,016	< 0.0001
2.4	0.608	2,850	0.4391
2.5	0.587	373	-----
3	**0.690**	**11,015**	
3.1	0.733	4,799	< 0.0001
3.2	0.637	796	0.3058
3.3	0.611	674	0.3582
3.4	0.590	1,433	0.5120
3.5	0.567	233	------
4	**0.530**	**52,731**	
4.1	0.617	19,044	< 0.0001
4.2	0.551	2,321	0.5579
4.3	0.561	1,336	< 0.0001
4.4	0.434	15,452	< 0.0001
4.5	0.380	3,263	------
5	**0.340**	**13,989**	
5.1	0.540	2,155	0.8180
5.2	0.548	228	0.2658
5.3	0.491	165	< 0.0001
5.4	0.332	3,165	< 0.0001
5.5	0.167	4,429	-----

2.3 Impact of opportunity count on 2SB combination predictions (Method 3)

The data set we are analyzing has been limited to only up to opportunity counts of 20. (This was done to speed the analyses.) Even with 25 two-SuperBin combinations, there was enough information in the data set to run a linear regression on the effect of when a two-SuperBin combination was reached. E.g. there is a difference between students who reach 1.1 (two right in a row) on opportunity 2 versus opportunity 20.

To create this model, pivot tables in excel were used to find the average next problem correctness (NPC) on two-SuperBin combinations that fall on particular opportunities. Although not all two-SuperBin combinations were achieved on all opportunities, there was enough information to run a linear regression. This, of course, gives an intercept and slope. The model was applied using the regression, not by using the actual calculated values.

2.4 Impact of 3 SuperBin (3-SB) combinations (Method 4)

Just as the state of the previous SuperBin could have an effect on future performance, it is conceivable that the SuperBin two opportunities back could have an effect. Consider the following two hypothetical students and their first three SuperBins:

Table 7: Two hypothetical students and their SuperBin values on three questions.

Student	Q1	Q2	Q3	Q4
Alice	SB 2	SB 1	SB 1	?
Barney	SB 5	SB 1	SB 1	?

Intuitively, we would expect Alice to have a higher probability of success on question 4 than Barney. Alice almost got the first question right, while Barney needed to use the bottom out hint (and used a hint as his first action). Although they both got questions 2 and 3 correct, their performances on question 1 are drastically different. To user models such as KT and PFA, however, they were both equally "wrong" on question 1.

To identify a 3-SuperBin combination, we will use the two-SuperBin code and add a decimal, we would have (current).(previous)(n-2) or [1.11-5.55]; this gives 125 three-SuperBin combinations. In the example above, after question 3 (and as the model predicts their correctness on question 4), Alice would be in 1.12, while Barney is in 1.15.

We are now looking at 125 combinations; some of these combinations have too few instances to have a prediction value that is reliable. 47 out of 125 3-SB combinations have fewer than 100 instances; eight combinations have 10 or fewer instances. Instead of using 125 different values (many of which would unreliable), we will use a linear regression to approximate values for the impact of the (n-2) SuperBin. However, it is a slightly complex process.

In order to have "smooth" regressions, some assumptions are made:

1.) The effects can be modelled linearly. E.g., for the regression to the (n-1) SuperBin prediction = intercept + slope*SuperBin (n-1).
2.) The effect of the (n-2) SuperBin value will be similar in pattern to the effect of SuperBin (n-1), but reduced in effect. (In other words, we would expect that 1.1x to follow the basic pattern of 1.x, but with a smaller change in values)
3.) Even though many of the three-SuperBin combinations are unreliable due to small numbers of instances, the average slope of a "family" could be used to deduce the effect size that is applied to the pattern found in assumption 2.

To create the model, five regression lines (one each for 1.x, 2.x, 3.x, 4.x, and 5.x) were created by simply using the average next problem correctness as the y-values and the decimal (previous SuperBin) as the x-values.

Next, twenty-five regressions were run for 1.1x - 5.5x. Although many of the three-SuperBin combinations were too small to be reliable, we used the average slope from a "family" (e.g. 1.3x) to adjust the effect from the two-SuperBin combination regressions. E.g., the regression lines for 1.1x - 1.5x were found and averaged.

To approximate the slopes of 1.1x-1.5x, the slopes of 1.x - 5.x were used, but multiplied by the ratio of the average (1.1x-1.5x) to the average (1.x - 5.x). Since the intercepts from (1.x-5.x) might not have the same meaning when compared to (1.1x - 1.5x), the intercepts from the three-bin regressions were left as is. Table 8 (below) shows the 2-SuperBin regressions (found using the values in Table 6), followed by the actual regression values for one of the 3-SuperBin families, and the idealized slopes.

2.5 First Possible Opportunity Count

Lastly, when fitting our methods (many of which would have to be some combination of the above four versions), we decided to separate SuperBins and combinations by the first available opportunity count, and all others. In our numbering scheme, we used a "dummy code" of 09 to designate that we are looking at the average of NPC for only the first available opportunity count. See next section for examples which may help.

2.6 Method Examples

We now arrive at the methods by which our model is applied. To see the differences between the methods, it may be useful to look at the same hypothetical sequence of SuperBins for two imaginary students and compare the different methods. (See Table 9, next page.) In all methods below, we compare "Chuck" and "Denise" and the parameters that would be used to predict their success. It's important to note that method 1 identifies the SuperBin into which each student is placed on questions 1-4; this does not change throughout the methods.

The simplest method uses only the average NPC for all SuperBins, and pays no attention to opportunity count or SuperBin combinations. This is Method 1. This can be thought of as a simplified FGA.

The prediction for (e.g.) question 5 is based solely on the SuperBin value for question 4. SuperBin values are modified by a multinomial logistic regression based on skill. This gives a total number of parameters as 5 + 1/skill.

In method 2, the prediction of NPC for question 1 is based on the average value for the SuperBin, but only including values from the first opportunities. (The "dummy code" of 09 is used to indicate first opportunity only.) All questions from then on use the value for the two-bin combinations. This gives a total number of parameters of 30 + 1/skill. (Five for SBx.09, and 25 for 1.1-5.5, plus the regression to skill)

In method 3, the prediction of NPC from question 1 is based on SuperBin at first opportunity, while all others are based on the regression to opportunity count values. This gives a total number of parameters of 55 + 1/skill. (Five for SB x.09, and 50 for the intercept and slope of the 25 different two-SuperBin combinations, plus the regression to skill)

In method 4, the prediction of NPC for question 1 and 2 are based on SuperBin x.09 and 2-SuperBin combinations x.y09. For question 3 and on, the prediction is based on the linear regression to the SuperBin of (n-2). This gives a total number of parameters of 65 + 1/skill. (Five for SB x.09, 25 for 2SB combo x.y09, 25 for the intercepts, five for the slope of 1.x-5.x, and five for the slope modification parameter, plus the regression to skill). The slope and intercept in the regressions in method 4 are not the same as those in method 3.

A demonstration of the application of all four methods can be found in Table 9 below. Method 1, being simply the single SuperBin prediction identifies a "score" or "condition" for the hypothetical students. The other methods start with this information.

Table 8: demonstration of idealization of regression to third bin using second bin regression values.

2 SB "family"	m	b
1.x	-0.047	0.898
2.x	-0.048	0.818
3.x	-0.038	0.741
4.x	-0.059	0.686
5.x	-0.096	0.704
3 SB "family" actual	m actual	b actual
1.1x	-0.032	0.913
1.2x	-0.031	0.845
1.3x	-0.025	0.089
1.4x	-0.034	0.778
1.5x	-0.018	0.695
3 SB "family" idealized	m idealized	b actual
1.1x	-0.023	0.913
1.2x	-0.023	0.845
1.3x	-0.018	0.089
1.4x	-0.029	0.778
1.5x	-0.047	0.695

Table 9: Hypothetical application of four different methods; it is important to note that the methods are different, but the results of "Chuck" and "Denise" are not. "X.09" (or "X.Y09") is a code meaning prediction values are derived from the first available bin only. E.g. "1.09" uses only the scores from SuperBin 1 and the first opportunity.

Method 1: SuperBin Only ("SB_1")					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB1	SB2	SB1	SB1	...
Denise	SB5	SB3	SB1	SB2	...
Method 2: Two-SuperBin combinations ("SB_2")					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	2SB (2.1)	2SB (1.2)	2SB (1.1)	...
Denise	SB 5.09	2SB (3.5)	2SB (1.3)	2SB (2.1)	...
Method 3: Two-SuperBin combinations, with opportunity regression ("SB_3")					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	b(2.1) +m(2.1)*2	b(1.2) + m(1.2)*3	b(1.1) + m(1.1)*4	...
Denise	SB 5.09	b(3.5) +m(3.5)*2	b(1.3) + m(1.3)*3	b(2.1) + m(2.1)*4	...
Method 4: Two-SuperBin combinations, with third bin regression ("SB_4")					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	2SB (2.109)	b'(1.2) + m'(1.2)*1	b'(1.1) + m'(1.1)*2	...
Denise	SB 5.09	2SB (3.509)	b'(1.3) + m'(1.3)*5	b'(2.1) + m'(2.1)*3	...

3. RESULTS

In order to better show methods, many of the tables that would be considered "results" are found throughout the paper. We hope this does not inconvenience the reader too much at this time.

Tables 1 through 5 show that a statistical analysis of student actions (based on next problem correctness) can simplify a complex table, while still retaining meaningful groupings of student actions. In our last paper, we argued that not only should hint use and attempt count be used in the model, but a simple action-order analysis should be included. We can point out that the regrouping process does not contradict this conclusion. Had group A not been split from group B, the model might not have fared so well.

Table 6 demonstrates that there is an effect of the previous SuperBin that will modify the prediction of the current SuperBin. For example, we can see that students in SuperBin 1 who were just in SuperBin 5 have almost a 20% (absolute) less chance of success on the next problem when compared to a student who was in SuperBin 1 twice running. This may not be too surprising, as SuperBin 1 represents getting the answer right. However, there is still a roughly 15% (absolute) difference in expected outcomes between 2SB 1.2 and 1.5. Both of these represent a student who got a problem wrong, and then got the next right. Algorithms such as KT or PFA would treat these conditions as identical.

When analyzing the 2SB combinations, the pattern is amazingly clear: the impact of wrongness does not disappear after one question, and the different levels have different (and predictable) impacts. Being in SuperBin 5 on the previous problem gives a student a worse outcome than 4; 4 is worse than 3, etc. There are only a few deviations from this pattern throughout Table 6. The p-value analysis indicates that the differences are reliable most of the time; that is, the patterns appear to be reliable, although a larger dataset is needed to state that definitively across all patterns.

One interpretation of the pattern of effect from the previous SuperBin would be that students in SuperBin 5 have more to learn than those in SuperBin 4, and that even getting the next question right is not a clear sign of having learned the knowledge component. The summary table (Table 5) gives another interpretation on this: the students in SuperBin 5 needed a hint before they even got started, and then needed the answer to finish. Clearly, these students are nowhere in the same state of learning as a student who makes one mistake and fixes their answer on their own (SB2).

This differentiation of "wrongness" demonstrates the power of looking at non-binary correctness. Perhaps the most dramatic observation is that a student who is wrong twice, but corrects themselves each time (2SB combination 2.2) is very different from a student who cannot start without a hint and cannot get to the correct answer on their own (2SB combination 5.5). To treat these two states as the same (wrong twice running) is to give up on information that can help differentiate a student who is nearly 70% likely to be correct on the next problem, verses one who as a paltry 16.7% chance (2.2 vs 5.5).

With these new predictions, we can compare predictions to other models. In Table 10, we compare the scores from RMSE, AUC, and R-squared. This shows that not only is the "SuperBin" method as valid as the FGA model (tying in two out of three metrics), taking opportunity regression (method 3) and 3-SB regression both improve on the basic SuperBins idea (method 1).

One table that a reader might be missing is one detailing the relation of 2SB to opportunity. Rather than add an eleventh table, we will summarize as: the R^2-values for the regressions ranged from 0.832 to 0.001; some are clearly not reliable. However, given the results in Table 10, we think that accounting for opportunity count by linear regression to the 2SB combinations is a worthwhile first approximation.

Table 10: Analysis of various knowledge models. Baseline predicts the average value of the training set. For AUC, 1.0 is ideal; 0.5 is no better than random. RMSE: 0.00 is ideal; 0.5 is no better than random. R^2: 1.00 is ideal, 0.0 is no better than random.

Method	AUC	RMSE	Rsqr
Baseline (predict mean)	0.500	0.446	0.000
PFA [11]	0.653	0.426	0.058
KT [6, 4, 10]	0.710	0.413	0.115
SOA [7, 17]	0.708	0.426	0.087
AM [15]	0.714	0.422	0.103
FGA [13]	0.715	0.400	0.128
SB method 1	0.715	0.411	0.128
SB method 3	0.726	0.407	0.142
SB method 4	0.727	0.406	0.145
Avg (methods 3 & 4)	0.728	0.406	0.145

4. CONCLUSIONS

The regrouping of 16 bins of the FGA into 5 "Super Bins" does not adversely affect the predictive power of the model (in two out of three metrics). In fact, by having fewer bins, we are able to look at history in a way we would not have, had we kept the 16 bins of the Fine-Grain Action model. This gives us a chance to improve on the FGA.

We can conclude that not all wrong answers[2] are equal, and that there is value to be gleaned from analyzing different wrong answers. The impact of "how wrong" an answer is has an effect even up to two answers later. That is, your "wrongness" two questions back can be used to make a better prediction for your next problem. (It is possible that wrongness further back could be used, but it would require a dataset that is larger by orders of magnitude.)

Not only is the combination of "wrongness" useful in making predictions, so too is the opportunity on which a student achieves a combination. That is, a student who gets the first two questions right is (usually) more likely to get the third right than a student who gets the 11^{th} and 12^{th} questions right is to get the 13^{th} correct.

It is perhaps not too surprising that this method is able to outperform established models such as PFA and KT. (And we will freely admit that the previous statement is limited only to this one dataset; more research is needed to definitively make this statement.) PFA and KT use only the information in binary correctness. A new model that outperforms existing models by using additional information does not negate the previous models; it merely shows that this information is worth incorporating into models of user knowledge.

[2] Or, more precisely, combinations of student actions that are treated as wrong answers; actual analysis of wrong answers is left to another paper.

4.1 Answers to the Research Questions

1.) The bins from FGA were useful, but needed to be regrouped. Regrouping by next problem correctness (and t-test analysis) kept local and logical groupings that yield meaningful descriptions of wrongness.

2.) The level of wrongness that a student demonstrates has an effect on more than just the current question. This effect is clear and reliable on the next problem and may impact the following.

3.) Not all wrong answers are identical. Knowledge estimation models such as KT and PFA leave out "levels" of wrongness that can be used to make a more accurate prediction of student success.

The paraphrased Anna Karenina quote at the start of the paper summarizes both our hypothesis and our findings: A careful analysis of wrong answers will help improve knowledge estimation models.

4.2 Novel Contributions

This paper seeks to show that there is information to be gained by treating different kinds of wrong answers as different. Presented herein is a statistical method of differentiating student actions into groups of actions that represent meaningful differences in performance. Use of these groups in a knowledge modelling algorithm can improve the results of the predictions, without needing continuous values (as in [14]).

4.3 Future Work

Although all of the linear regressions can be considered first-order approximations, the idealization of the third bins may be perhaps only a zeroth-order. As more data becomes available, we may be able to bypass the idealization and simply use 125 different parameters that are statistically reliable. Beyond improving the results of this model, the incorporation of other models that seek to use information from incorrect answers should bolster the performance of the model(s).

5. ACKNOWLEDGEMENTS

Special thanks also go out to the ASSISTments team for all the work they do in harvesting the data from the system. We also acknowledge and thank funding for ASSISTments from the NSF (1316736, 1252297, 1109483, 1031398, 0742503, and 1440753), the U.S. Dept. of Ed. GAANN (P200A120238), ONR's "STEM Grand Challenges," and IES (R305A120125, R305C100024).

6. REFERENCES

[1] Attali, Y., & Powers, D. (2010). Immediate feedback and opportunity to revise answers to open-ended questions. *Educational and Psychological Measurement*, *70*(1), 22-35.

[2] Baker, R. S., Goldstein, A. B., & Heffernan, N. T. (2010). Detecting the moment of learning. *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

[3] Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *Affective Computing, IEEE Transactions on*, *1*(1), 18-37.

[4] Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006). A bayes net toolkit for student modeling in intelligent tutoring systems. Intelligent Tutoring Systems. Springer Berlin Heidelberg.

[5] Campione, J. C., Brown, A. L., & Bryant, N. R. (1985). Individual differences in learning and memory. In R. J. Sternberg (Ed.). *Human abilities: An information-processing approach*, New York: W. H. Freeman. pp. 103-126.

[6] Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4), 253-278.

[7] Duong, H. D., Zhu, L., Wang, Y., & Heffernan, N. T. (2013). A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for a Hint. *Proceedings of the 6th International Conference on Educational Data Mining.* 2013.

[8] Grigorenko, E. L., & Sternberg, R. J. (1998). Dynamic Testing. Psychological Bulletin, 124, 75-111.

[9] Hawkins, W., Heffernan, N., Wang, Y., & Baker, R. S. Extending the Assistance Model: Analyzing the Use of Assistance over Time.

[10] Murphy, K.: Bayes Net Toolbox for Matlab. < https://code.google.com/p/bnt/ > Accessed 4 September, 2014

[11] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.

[12] Sternberg, R.J., & Grigorenko, E.L. (2002). Dynamic testing: The nature and measurement of learning potential. Cambridge, England: Cambridge University Press.

[13] Van Inwegen, E. G., Adjei, S., Wang, Y., & Heffernan, N. T. An Analysis of the Impact of Action Order on Future Performance: the Fine-Grain Action Model, LAK2015, in publication

[14] Wang, Y., & Heffernan, N. (2013). Extending knowledge tracing to allow partial credit: using continuous versus binary nodes. *Artificial Intelligence in Education*. Springer Berlin Heidelberg.

[15] Wang, Y., & Heffernan, N. T. (2011). The" Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *FLAIRS Conference*.

[16] Wang, Y., Heffernan, N. T., & Beck, J. E. (2010). Representing Student Performance with Partial Credit. *EDM*.

[17] Zhu, L., Wang, Y., & Heffernan, N. T. The Sequence of Action Model: Leveraging the Sequence of Attempts and Hints.

Translating Head Motion into Attention - Towards Processing of Student's Body-Language

Mirko Raca
CHILI Laboratory
École polytechnique fédérale de Lausanne
RLC D1 740, CH-1015 Lausanne
mirko.raca@epfl.ch

Łukasz Kidziński
CHILI Laboratory
École polytechnique fédérale de Lausanne
RLC D1 740, CH-1015 Lausanne
lukasz.kidzinski@epfl.ch

Pierre Dillenbourg
CHILI Laboratory
École polytechnique fédérale de Lausanne
RLC D1 740, CH-1015 Lausanne
pierre.dillenbourg@epfl.ch

ABSTRACT

Evidence has shown that student's attention is a crucial factor for engagement and learning gain. Although it can be accurately assessed ad-hoc by an experienced teacher, continuous contact with all students in a large class is difficult to maintain and requires training for novice practitioners. We continue our previous work on investigating unobtrusive measures of body-language in order to predict student's attention during the class, and provide teachers with a support system to help them to "scale-up" to a large class.

Our work here is focused on head-motion, by which we aim to mimic large-scale gaze tracking. By using new computer vision techniques we are able to extract head poses of all students in the video-stream from the class. After defining several measures about head motion, we checked their significance and attempted to demonstrate their value by fitting a mixture model and training support vector machines (SVM) classifiers. We show that drops in attention are reflected in a decreased intensity of head movement. We were also able to reach 61.86% correct classifications of student attention on a 3-point scale.

Keywords

computer vision, head movement, attention, classroom

1. INTRODUCTION

One of the early studies of attention in classrooms showed that only 46% of students pay attention during the class [4]. Later studies raised that estimation to a more optimistic but still insufficient 67% [20]. This means that in practice the teachers are lecturing half-empty classrooms, even if all chairs are occupied. How can we help the teachers learn to recognize which chairs are empty?

Processing of social cues comes natural in human-to-human communication, but still remains an object of much research and few technical applications. The ambiguity of the medium limits our attempts, but in the scenarios where body language becomes the dominant form of expression, we are inclined to dig further into the matter. One such scenario is the classroom. We argue that computer vision (CV) technologies, in combination with machine learning approaches give us tools to scale-up teacher's attention to every student in the classroom, regardless of the class size. This would provide the teachers with a timely opportunity to address lower attentive class areas and draw students into the lecture, encouraging teacher's reflection in action.

Behaviour of people in large groups is unpredictable to an observer in most situations. The overwhelming amount of information forces us to focus on few individuals who we deem as the representatives of the group, and mental effort and training are required to re-divide the attention equally among many subjects [7]. In case of a lecture, teachers are active participants, splitting their attention between personal actions, material presentation and orchestration of the whole process [8].

In this work we started from the success of eye-tracking in predicting focus and tried to generalize it to students' head movement in the classroom. Birmingham et al [3] illustrate the social aspect of gaze – given an image, people first analyse the gaze, then the head and finally the posture of the people in the image to collect information about where to focus their attention. Langton [13] showed that we combine the input from head and eyes into a single stimulus. These two observations together gave us the ground to consider head orientation as *i*) informative to other humans, and thus potentially also for our algorithms; *ii*) an approximation of human gaze on larger scales of motion.

In this paper we present our process for extracting head motion and pose features from videos of classroom audience, and our initial set of analysis of the features' quality. We will try to answer if there is a general connection between head motion and attention level? What are the features of head motion that we can use in predicting attention? How do these features change with attention levels? And finally, can we use these features to predict students attention levels?

2. RELATED WORK

The umbrella of affective computing [15] has been growing in the last 15 years, and expanding the domains of it's application. The emerging sub-field of Social Signal Processing (SSP) [24, 25] made a major point of emphasizing that encoding human social and cultural information might raise the performance of the machine algorithms aimed at understanding behaviour (e.g. analysing large sport gathering [6]).

In case of human attention, it is attributed with the ability to modulate or enhance the selected information source according to the state and goals of the perceiver, and that the "perceiver becomes an active seeker and processor of information, able to intelligently interact with their environment" [5] and can be highly relevant in a learning environment [14]. Roda et al [19] already tried to incorporate the attention indication as one of the inputs in human-computer interaction, but early attempts in the classroom were not formulated as a technology which can be wide-spread, due to their complexity [1].

Detecting and displaying the gaze direction, as one of the key indicators of focus of attention, was shown to be both useful in making the interaction feel more natural [23], and indicative of the material comprehension [21] in on-line environments. Lacking the possibility of capturing gaze in a real-life scenario, Ba et al [2] demonstrated that we can estimate the VFOA (visual focus of attention) in meetings successfully based on the head pose. In the similar scenario Stiefelhagen et al [22] showed that head orientation contributes 68.9% in the overall gaze direction (where is the attention directed) and achieved 88.7% accuracy at determining the focus of attention. This gives us the indication that head motion has potential as a focus indicator, but it does not come without problems. Deeper exploration of head motion depicts it as an ambiguous indicator. Heylen's overview [10] shows that head-signals are either very contextual-dependant or are complementary signal to the main information channel (usually – talking).

Our conclusion from the literature overview is that head motion has the potential as a low-resolution measurement which we can passively acquire to determine the attention level and/or direction of another person. To fully decode it we need contextual information which will be unavailable in our approach of passive/unobtrusive data collection [16]. The features we hope to find need to be positioned in the middle between measurable and context-dependant.

3. METHOD

Training and validation of our head detector/pose estimator pipeline was detailed in our previous work [17]. We will give a quick overview of the experiment setup and detection pipeline, and focus on the steps and problems we encountered in the later stages of data extraction.

3.1 Experiment design

We collected a total of 6 recorded sessions with 2 classes (demographic information shown in Table 1). Each classroom was observed with several cameras positioned above teacher's head around the blackboard area of the classroom (camera view of the classroom is shown in Figure 1). The

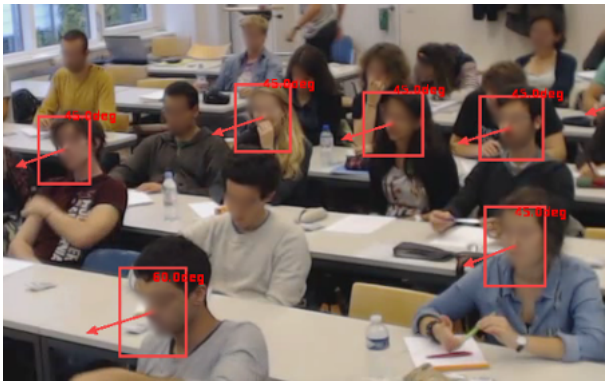


Figure 1: Examples of gaze detections, showing the classroom during the lecture.

cameras were synchronized and each student visible in the video was annotated with an unique ID (maintained over all recorded sessions) and a rectangular area of the video which the student occupies. Given that the angle of the face detected is relative to the camera viewpoint, we introduced angle offsets for each student. If a student was visible from several cameras, best quality recording was used.

Class	Size	F.ratio	Mean attend.	Sess	Cams
1	62	35.48%	39.34($\sigma = 1.15$)	3	5
2	43	34.88%	27.5($\sigma = 6.55$)	3	4

Table 1: Statistics of the two captured classes, showing the number of students, percentage of female students, attendance, number of sessions recorded and number of cameras used.

Similar to attention probing used in earlier experiments [4] we asked students to fill out the questionnaire about their attention during the class. At four different times the classes were interrupted and students recorded their attention on a Likert scale from 1–10 (details of the questionnaire design are presented in [17]). The distribution of all collected answers is shown in Figure 2. From each of the 6 processed classes we recorded 4 measurements of attention per student, associated to the time period before our interruption, duration of 7-10 minutes. In order to turn the problem into a classification one, we labelled the values of the students' responses as *low* (reported attention 1–4), *medium* (5–7) or *high attention* (8–10), based on our observations of attention distribution (regions marked in Fig.2).

3.2 Video analysis

The head-pose detection and pose estimation was built on top of the part-based model for head detection published by Zhu et al [26] which was re-trained for lower resolution images and different head poses on the AFLW dataset [12]. We trained a geometrical head-pose estimator (focusing on horizontal angle or "*pan*" of the head) by using the dlib library [11]. The precision of the estimators was checked on the Pointing'04 dataset [9]. Each detection consists of the assumed rectangle of face area, estimated angle of the face ("pan") and score (detector confidence).

The major problem for reaching the meaningful measure-

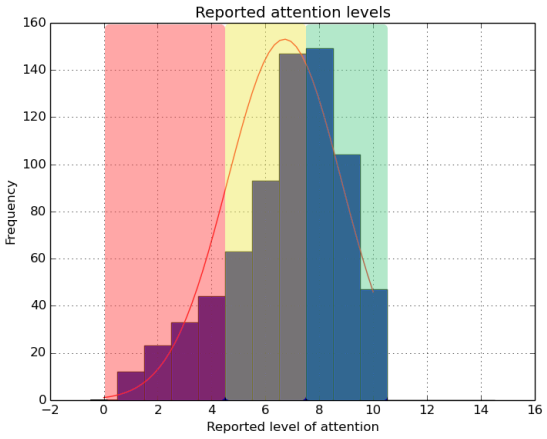


Figure 2: Histogram of all reported levels of attention with the used limits to designate the *low* (red zone, <5), *medium* (yellow 5-7) and *high* (green, 8-10) levels of attention.

ments was the instability of the detector/estimator output. The measurements were very noisy since the feature extraction step was not formulated as a tracker, which would provide temporal consistency. The second problem came from the setup itself — given the location of the cameras (around the black-board, visible in Figure 1), the subjects sit closely together. This causes a considerable amount of *i*) interpersonal occlusions and *ii*) gaps in detection and *iii*) miss-assignment of detection instances (visualized in Figure 3a).

Simple attempts to pick the best-scoring detection within the region did not yield a stable output, given that on most occasions the head of the neighbouring student would wander into the region and take over as the best detection. Fitting prior distributions (2D Gaussians) for expected head locations also did not improve the assignment, as students usually create 2 or 3 mixtures of points (depending on their sitting poses), which is indistinguishable from the case when two people occupy the given space.

Finally we settled for the formulation with labelled GMM (Gaussian Mixture Model). By taking sparsely sampled detections over time (one frame every 2 seconds) and accumulating all the detections, we depicted the overall probability of detecting faces in different positions of the camera view. The "labelled" part consists of manually specifying the relevance of each mixture in the probability, by either labelling the mixture as a specific person or miss-detection. With this we could filter-out all the irrelevant detections for a specific person by only considering detections which were assigned to one of the person-related clusters in the GMM (Figure 3b).

To improve the precision of the GMM fits, before training the model we eliminated the outlier points by thresholding the minimal number of neighbours a point needs to have in order for it to be further considered. This is possible due to the fact that the people remain in distinct positions for long periods of time, causing dense groupings of detections. The threshold was dynamically determined for each video, by eliminating the 0.5% of points with lowest number of neighbours. The major role of the GMM filtering step was to eliminate false positives, as the clusters could not always be mapped one-to-one to an individual. Additional constraints during the GMM training phase could solve this problem.

After filtering out the miss-detections, temporal consistency was ensured by using a simplified Kalman filter approach – the next detection is expected to be in the close proximity of the previous detection. If no detections were observed within a specified radius from the previous detection, the radius is increased for the next processed frame and no detection is reported, simulating the increase in uncertainty. The major differences from the Kalman filter is the absence of motion model (the face is expected to remain at the same place) and the lack of probability propagation. This enabled us to use only the real detections and not estimates, which is relevant in order to model the heads in a bow-down position. The region growing was preferred over moving Gaussian in order to put a hard limit on the detections which can be considered.

After each processed person in the video, to make sure that the detection would not be used two times, we removed the detection after it has been assigned to a person. This turns the algorithm into a greedy approach, and making the order in which the persons are processed important. We chose to process the persons from front-to-back given that each person sitting closer to the cameras is more likely to be correctly detected. After extracting detection tracks for each person, values of the detection rectangle position and gaze angle are smoothed with a "sliding window" approach.

3.3 Features extracted

The input features used in our predictions were largely based on the information extracted from the cameras, but not exclusively. All features used are shown in Table 3.3. As we noted before, the time and spatial arrangement also plays significant role in the attention estimation [18], so we included the information about the distance of the student from the teacher (distance and row fields), and time of the sample within the class (period).

We tried to model the eye contact in the class with the percentage of time that we detected the student's face in the video. Initial assumption is that this would allow us to measure the time the student spent looking down just by noting how long was the head absent. The noise in the measurement originates from the false negatives of the detector, which is dominantly influence by the distance from the camera. Even though we resorted to using zoom-lenses for the distant people in the class (which makes the measurements comparable even on the capture level to the people in the front rows), there still was a significant correlation between the row in which the student sat and percentage of time detected ($r = -0.1867$, $p = 0.009$), although it was weaker than the correlation with the Cartesian distance from the teacher ($r = -0.2137$, $p = 0.002$) which encodes width as well as depth of the classroom.

"Head travel" records the total accumulated head travel in the horizontal plane. We ignored the potential head-travel in the periods when we did not detect the face of the stu-

dent. In order to neutralize the potential influences of person's rhythm and distance from camera, we also included a normalized version of the measure, by using all the measurements of a single person to determine the mean and scaled it with the variance of those measurements. Samples with a single measurement were excluded.

We modelled the focus of the student with 3 connected measures of stillness – number of still periods, mean duration of the still period and percentage of time spent still. Stillness was defined as periods during which the head changes are less than 10°, and where the head's angle does not move away from the initial angle more than 10° (in order to prevent slow drifting to be classified as stillness). "Stillness periods" are defined as non-overlapping periods of minimum duration of 5 seconds, in which the stillness condition is true. From there we get the first two measures by counting the number of such periods and their mean duration. Percentage of time spent still is the ratio of time classified as being still over the duration of the attention period.

All measurements were considered per attention period and per person in order to associate the features to the labels acquired from the questionnaire. In case of regressions/ correlation tests, we also tested the correlation of the measures after the logit transformation, by first bounding the value scopes (finding minimum and maximum values for all measurements and scaling them to the 0.1 – 0.9 interval) and applying the $\log_e\left(\frac{p}{1-p}\right)$.

4. RESULTS AND DISCUSSION

4.1 Features

First significance tests showed the correlation between the pure attention level with the percent of time the person was detected (Pearson's $r = 0.1158$, $p = 0.01$, 577 samples). This can be explained with the idea that engaged students will maintain more contact with the activities in the classroom. Apart from being more visible, students head travel did not show significant difference on the overall scale. We expected this as the measurement itself can be easily affected by noisy measurements, even though we did take steps in smoothing the data.

Head travel became significant when testing its potential to measure the change in behaviour. After eliminating the individual differences with normalization of head travel, we found that positive changes in attention were reflected in increase in head travel (Pearson's $r = 0.21$, $p < 0.01$, 236 samples), as shown in Figure 4.

Of the measures of stillness, only "percentage of time spent still" recorded a significant, but very weak correlation (Pearson's $r = 0.09$, $p = 0.02$). After comparing it with the "percentage of time detected" we found a very high and significant correlation between the two measures ($r = 0.91$, $p < 0.01$), which does not allow for great significance of the measure. We kept the measures for further testing.

4.2 Models

Next step in demonstrating the usefulness of the features was to try to predict the attention levels based on their combinations. After initial attempts with linear regression

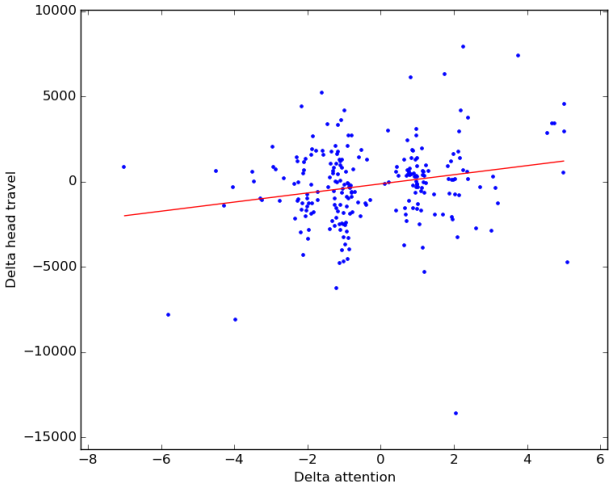


Figure 4: Change in normalized head travel correlated to the change in attention. Red line represents the linear fit. Pearson's $r = 0.21$, $p < 0.01$. Number of samples 236. Noise added for the visualization after the linear fit.

which were not successful, we switched to the mixture model. Our mixed model for *logit attention* (**A**) with *period* (**P**), *row* (**R**), *number of still periods* (**N**) and *head travel normalized* (**H**) takes form

$$L(A) = 1.061 - 0.060P - 0.128R + 0.012N - 0.035H.$$

Although its predictive power ($R^2_{random} = 0.54$ and $R^2_{fixed} = 0.05$) is limited, significance encourages further investigation of more advance supervised learning methods.

With that in mind, we tried an exhaustive search of all feature combinations and SVM parameters to achieve the best prediction of the three categories of "labelled attention" – *low* (100 samples), *medium* (270 samples), *high* (246 samples). Training of the classifiers was repeated in several rounds (500 iterations) with random drawing of training and testing samples, while making sure that the ratio of samples for each output category is maintained (roughly 16%, 44% and 40%). Our training procedure was based on the 80–20 split — 80% of the data used for training, and 20% data for testing the prediction of the trained classifier. To evaluate SVM parameters during the training we additionally split the 80% used for training into another 80–20 split. This gives us the final data configuration — 64–16–20 split, where 64% of the data was used for training, 16% for evaluating the SVM parameters during the training, 20% for the final evaluation of the trained classifier.

For each combination of features we iterated over the SVM parameters with sampling step of 0.1 (kernel type considered - *linear, polynomial, rbf*, and their relevant parameters). On the top scoring feature combinations we applied gradual refinement of the parameter sampling step (step size was reduced down in sequence 0.1, 0.01, 0.001 around the best scoring parameter values from the previous round). Four best scoring classifiers are given in Table 3, with the best result of 61.86% correct classifications (Cohen's kappa 0.30)

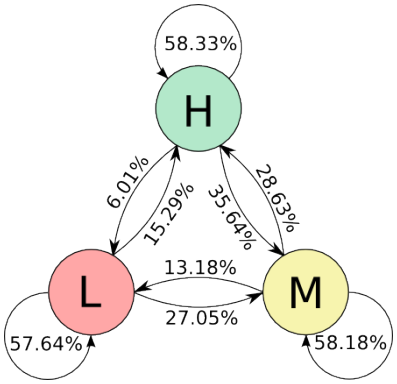


Figure 5: Transition probabilities between the three attention levels (*low, medium, high*).

on the independent test set.

Our concern was that the main informative source would rely on the *Detection percentage* or *Percentage still*, the two being highly correlated. This did happen in the early training attempts, but the features are not represented in the final set of classifiers (*Detection percentage* is used in the 10th best classifier). All of the best classifiers included a similar mix of features – head motion representatives, and some indications of distance and time of the class. *Normalized head-travel measurements* and *Mean duration of still periods* appears to be the most salient feature (both used in 3 of the 4 detectors).

Even though we saw no significant correlation of attention with class period in the feature analysis, we also tested the "attention labelled" for Markov property and got highly informative transitions probabilities shown in Figure 4.2. The trend of remaining in the same state with lower possibilities of transition to neighbouring, although not directly relevant to the attention level definitely puts additional constraints on the predictions. In order integrate this knowledge into our model, the next step was to connect our SVM predictions (observational model) and temporal consistency (transition probabilities) into a Hidden Markov Model, but due to time constraints we are unable to report the results in this publication.

5. CONCLUSION

The goal of this study was not only to answer questions about the link between student's movement and attention, but also to investigate to what extent can we approximate these variables by current techniques, without manual annotation. We defined a number of head metrics that can be extracted from a video of the audience attending a class. Considering measures that are "global" in nature (not relying on specific events such as gesturing, nodding etc.) we have shown that the change in head motion usage correlates with the change in reported level of attention. We also experimentally confirmed that higher percentage of head detection mirrors higher time spent in contact with the classroom events, indicating higher attentiveness.

For classification tasks, we found that head measurements alone were not enough to give us definitive answers about the person's attention. Each of the high-scoring classifiers used other contextual cues which related person's actions to the temporal or spacial domain (e.g. class period, distance). Also, in this report we did not explore social-level cues – how the students actions are contrasted against their immediate environment or general classroom population. We have expectations that these features will provide further contextual information, which will raise the precision of predictions.

Apart from the "global" measurements, we are also looking to explore discrete gestures which can be detected with the system (e.g. nodding, yawning, turning), of which only "bowing the head down" was used at this stage, encoded within the "percentage of time detected". The problem that we perceive is that the noise of the measurements was evident in the current setup, and that relying on the features which are more sensitive will depend on further improvements in the computer vision algorithms.

Our current conclusion is that the technology shows promise and that future investigations will bring higher accuracy and new tools to the classrooms. Our future work will try to work in parallel on finding more meaningful measures, and coordinate with the teachers to determine the best way to present the found information back to the teaching process.

6. REFERENCES

[1] I. Arroyo, D. G. Cooper, W. Burleson, B. P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *AIED*, volume 200, pages 17–24, 2009.

[2] S. O. Ba and J.-M. Odobez. Recognizing visual focus of attention from head pose in natural meetings. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):16–33, 2009.

[3] E. Birmingham, W. F. Bischof, and A. Kingstone. Social attention and real-world scenes: The roles of action, competition and social content. *The Quarterly Journal of Experimental Psychology*, 61(7):986–998, 2008.

[4] P. Cameron and D. Giuntoli. Consciousness sampling in the college classroom or is anybody listening?. *Intellect*, 101(2343):63–4, 1972.

[5] M. M. Chun and J. M. Wolfe. Chapter nine visual attention. *Blackwell Handbook of Sensation and Perception*, pages 272–311, 2001.

[6] D. Conigliaro, F. Setti, C. Bassetti, R. Ferrario, and M. Cristani. Attento: Attention observed for automated spectator crowd analysis. In *Human Behavior Understanding*, pages 102–111. Springer, 2013.

[7] J. A. Daly and A. Suite. Classroom seating choice and teacher perceptions of students. *The Journal of Experimental Educational*, pages 64–69, 1981.

[8] P. Dillenbourg, G. Zufferey, H. Alavi, P. Jermann, S. Do-Lenhand, Q. Bonnard, S. Cuendet, and F. Kaplan. Classroom orchestration: The third circle of usability. In *International Conference on Computer Supported Collaborative Learning Proceedings*, pages 510–517. 9th International Conference on Computer Supported Collaborative Learning, 2011.

[9] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial structures. In *FG Net Workshop on Visual Observation of Deictic Gestures*, pages 1–9. FGnet (IST–2000–26434) Cambridge, UK, 2004.

[10] D. Heylen. Challenges ahead: head movements and other social acts during conversations. In L. Halle, P. Wallis, S. Woods, S. Marsella, C. Pelachaud, and D. Heylen, editors, *Joint Symposium on Virtual Social Agents*, pages 45–52. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2005. Imported from HMI.

[11] D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[12] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[13] S. R. Langton. The mutual influence of gaze and head orientation in the analysis of social attention direction. *The Quarterly Journal of Experimental Psychology: Section A*, 53(3):825–845, 2000.

[14] S. I. Lindquist and J. P. McLean. Daydreaming and its correlates in an educational environment. *Learning and Individual Differences*, 21(2):158–167, 2011.

[15] R. W. Picard. *Affective computing*. MIT press, 2000.

[16] M. Raca and P. Dillenbourg. System for assessing classroom attention. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 265–269. ACM, 2013.

[17] M. Raca and P. Dillenbourg. Holistic analysis of the classroom. In *Proceedings of the 2014 ACM workshop on Multimodal Learning Analytics Workshop and Grand Challenge*, pages 13–20. ACM, 2014.

[18] M. Raca, R. Tormey, and P. Dillenbourg. Sleepers' lag-study on motion and attention. In *Proceedins of the Fourth International Conference on Learning Analytics And Knowledge*, pages 36–43. ACM, 2014.

[19] C. Roda and J. Thomas. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, 22(4):557–587, 2006.

[20] J. R. Schoen. Use of consciousness sampling to study teaching methods. *The Journal of Educational Research*, 63(9):387–390, 1970.

[21] K. Sharma, P. Jermann, and P. Dillenbourg. "with-me-ness": A gaze-measure for students' attention in moocs. In *International Conference Of The Learning Sciences*, number epfl-conf-201918, 2014.

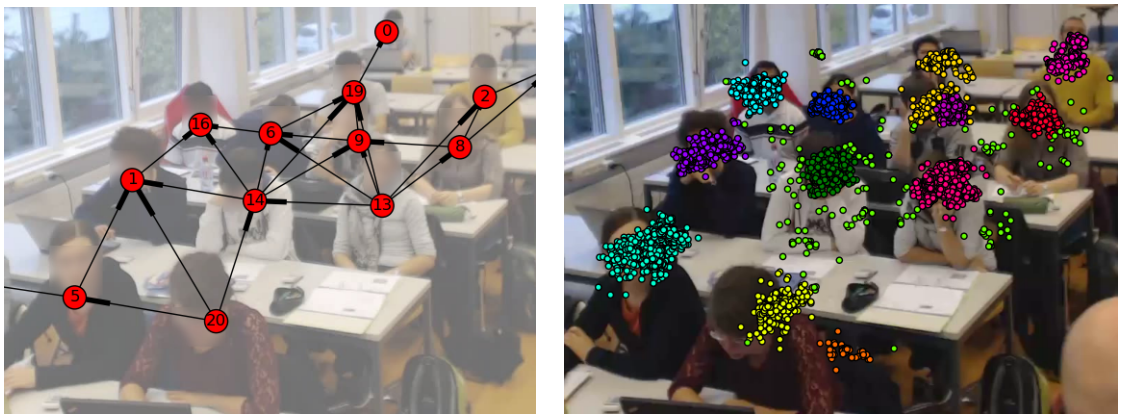
[22] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 858–859. ACM, 2002.

[23] R. Vertegaal. The gaze groupware system: mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 294–301. ACM, 1999.

[24] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.

[25] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D'Errico, and M. Schröder. Bridging the gap between social animal and unsocial machine: A survey of social signal processing. *Affective Computing, IEEE Transactions on*, 3(1):69–87, 2012.

[26] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.



a) b)

Figure 3: Processing of detections. *a*) Overlaps between subjects areas. Each graph edge shows neighbouring students areas and potential for miss-assignment of detections. *b*) All detections over the duration of the class, coloured depending on the cluster to which they were assigned.

Feature name	Description	Valid samples
Period	Period of the class (1–4), associated with the attention	776
Distance	Distance from the teacher on a Cartesian plane of the classroom	776
Row	Student's row in the classroom	776
Detection percentage	Percentage of the recorded time that the student was detected	668
Head travel	Accumulated changes (deltas) of the head horizontal rotations over time.	496
Head travel (norm.)	Head travel normalized over the measurements of the specific person in the class.	482
Number of still periods	Number of periods (of minimal duration of 5 seconds) during which the head movement can be considered still	668
Mean still period duration	Mean duration of the still period (as defined in the previous row)	618
Still time percentage	Percentage of time within the attention period during which the head was still.	668
Attention	Reported level of attention (1–10)	715
Attention labelled	Attention reports mapped to categories *low*, *medium*, *high*	715

Table 2: Features used in the analysis.

Kernel	Features	Score	Cohen's kappa
RBF(c=1.31, g=0.0211)	Distance, Head travel norm., Num. still periods	61.86%	0.30
RBF(c=1.21, g=0.11)	Period, Row, Head travel norm., Mean duration still	61.72%	0.32
RBF(c=1.11, g=0.061)	Head travel norm., Mean duration still	60.42%	0.28
RBF(c=1.4, g=0.04)	Period, Distance, Row, Mean duration still	59.23%	0.30

Table 3: Classifier scores for predicting "attention labelled". Score given represent the prediction score on the 20% test sample. Parameters of the kernels are abbreviated as *c* - penalty for the error term; *g* - gamma.

Using Visual Analytics Tool for Improving Data Comprehension

Jan Géryk
KD Lab Faculty of Informatics
Masaryk University
Brno, Czech Republic
geryk@fi.muni.cz

ABSTRACT

The efficacy of animated data visualizations in comparison with static data visualizations is still inconclusive. Some researches resulted that the failure to find out the benefits of animations may relate to the way how they are constructed and perceived. In this paper, we present visual analytics (VA) tool which makes use of enhanced animated data visualization methods. The time is an important variable that needs to be modeled in VA. VA methods like Motion Charts show changes over time by presenting animations in two-dimensional space and by changing element appearances. The tool is primarily designed for exploratory analysis of academic analytics and supports various interactive visualization methods which enhance the Motion Charts concept. We evaluate the usefulness and the general applicability of the designed tool with a controlled experiment to assess the efficacy of the described methods. To interpret the experiment results, we utilized one-way repeated measures ANOVA.

Keywords

Animation; motion charts; visual analytics; academic analytics; experiment.

1. INTRODUCTION

Higher education institutions have a strong interest in improving the quality and the efficacy of the education. In [1], hundreds of higher education executives were surveyed on their analytic needs. Authors resulted that the advanced analytics should support better decision-making, studying enrollment trends, and measuring student retention. They also pointed out that management commitment and staff skills are more important in deploying academic analytics (AA) than the technology. In [2], authors concluded that the increasing accountability requirements of educational institutions represent a key for unlocking the potentials of AA in order to effectively enhance student retention and increase graduation levels. The authors also resulted that AA facilitate creation of actionable intelligence to enhance learning and student success, however, it is highly dependent on the quality of the accountability. The authors utilized AA for developing several predictive models of student enrollment and retention, and for identifying students being at the risk. They also highlighted three critical success factors–executives committed to decision-making based on the evidence, staff members with adequate data analysis skills and the flexible and effective technology platform. However, the authors also warned that more elaborated accountability can raise several privacy issues, faculty executive's involvement, and data administration.

The principal goals can be achieved by using educational data mining methods, as emphasized in [3]. The application of data mining (DM) techniques in higher education systems have some specific requirements not present in other areas, as pointed out in [4]. Common DM methods were developed independently of visualization techniques. However, some key ideas influenced the research in the DM field. It resulted into the recent research topic called visual analytics (VA). Google Analytics, released in 2005, made a real progress in web-based interactive analytics. In 2007, Hans Rosling presented a TED talk demonstrating the power of animations to show the story in data. In 2009, Tim O'Reilly emphasized that data analysis, visualizations, and other techniques for searching patterns in data are going to be an increasingly valuable skill set [5]. While some researches resulted that animations appeared better than static visualizations in enhancing learning, an elaborate examination of the studies revealed a lack of equivalence between animated and static visualizations in content [6]. Also, the failure to ascertain the benefits of animations in learning may also relate to the way how they are constructed, perceived, and conceptualized [7].

Visualizations are common methods used to gain a qualitative understanding of data prior to any computational analysis. By displaying animated presentations of the data and providing analysts with interactive tools for manipulating the data, visualizations allow human pattern recognition skills to contribute to the analytic process. The most commonly used statistical visualization methods (e.g. line plots, or scatter plots) generally focus on univariate or bivariate data. The methods are usually used for tasks ranging from the exploration to the confirmation of models, including the presentation of the results. However, fewer methods are available for visualizing data with more than two dimensions (e.g. motion charts or parallel coordinates), as the logical mapping of the data dimension to the screen dimension cannot be directly applied. Data exploration and interactive visualizations of multivariate data without significant dimensionality reduction remains a challenge. Animations represent a promising approach to facilitate better perception of changing values. In [6], authors pointed out that animations help to keep the viewer's attention. Visualizations and animations can also facilitate the learning process [8].

We develop visualization methods for multivariate data analyses that are adapted for academic settings. In this paper, we show the importance of data visualizations for successful understanding of complex and large data. In the next section, we examine characteristics of changes using Motion Charts (MC). Subsequently, we present several papers successfully utilizing MC for data visualization and analysis. This is followed by the elaborate description of our VA tool. Further, we conducted an empirical study with 22 participants on their data comprehension to compare the efficacy of static and animated data visualizations. We then

discuss the implications of our experiment results. Finally, we draw the conclusion from the experiment and outline future work.

2. EXAMINE CHARACTERISTICS OF CHANGE

Although a snapshot of the data can be beneficial, presenting changes over time provides a more sophisticated perspective. The efficacy of animated transitions for common statistic data visualizations such as bar charts and scatter plots was examined in [9]. The authors extended the theoretical model of data visualizations and introduced the taxonomy of transition types. Subsequently, they proposed design principles for creating effective transitions and illustrated the application of these principles in a dynamic visual system. Finally, they conducted two controlled experiments to assess the efficacy of various transition types, finding that animated transitions can significantly improve the visual perception. The visualization challenge posed by each of these experiments was to keep the viewer's attention during transitions. The survey resulted that viewers found animations more helpful and engaging. Unlike transition animations, which primarily help users to stay in the context, trend animations convey the meaning. While a transition animation moves from a still view to a new still view, a trend animation moves continuously between states. One early use of animations in visualization was for an algorithm animation. Kehoe et al. [10] describe a study that demonstrated that animations could help and noted that it improved the motivation of making a difficult topic more approachable. The study suggested that using animations for trend understanding could be valuable.

Animations allow knowledge discovery in complex data and make it easier to see meaningful characteristics of changes over time. To reduce the cognitive load and improve tracking accuracy, the target states of all transitioning elements should be predictable after viewing a fraction of the animation. The proper use of the acceleration should also improve the spatial and temporal predictability. A perceptual study in [11] provides evidence that animations and divergence motions are easier to understand than rotations. Animations with unpredictable motion paths or multiple simultaneously changing elements result in the increased cognitive load. Contrarily, simple transitions reduce confusion and improve clarity. In [12], authors concluded that animation stages should be long enough for accurate change tracking as well as to decrease the number of errors. However, too slow animations can disproportionately prolong the analytic phase and subsequently reduce the engagement.

Generally, effective analyses depend on the consistent and high-quality data. In [9], authors concluded that the correctly designed animations significantly improve the visual perception at both the syntactic and the semantic level. Visualizations are often engaging and attractive, but a naive approach can confuse analysts. Visualizations are just representations of the data which may or may not represent the reality. As Few pointed out in [13], computers cannot make sense of the data, only people can. The perception of animations can also be problematic because of severe issues with timing and the overall complexity that can occur during transitions as pointed out in [14]. Misleading results can be obtained if animations violate the underlying data semantics.

MC is a dynamic and interactive visualization method that enables analysts to display complex and quantitative data in an intelligible way. The dynamic refers to the animation of rich multidimensional data changing over time. The interactive refers to dynamic interactive features which allow analysts to explore, interpret, and analyze information concealed in complex data, as presented in [15]. MC displays changes of element appearances over time by showing animations in a two-dimensional space. An element is basically a two-dimensional shape representing one object from the dataset. The variable mapping is one of the most important parts of the exploratory data analysis and no optimal method for mapping the data to variables is available. Naturally, the data mapping have a significant impact on the data comprehension and analysts should be free to choose variable mapping according to their intentions. Both the data characteristics and the investigative hypothesis influence the variable mapping.

3. APPLICATIONS OF MOTION CHARTS

Visualization tools represent an effective way how to make statistical data understandable to analysts, as showed in [16]. MC methods proved to be useful for data presentation and the approach was verified that can be successfully employed to show a story in data [17] or support decision making [18]. In [19], authors utilized MC for both the interpretation of results for better comprehension and the analysis when detecting topics of tweets. Several web-based data analysis tools allowing analysts to interactively explore associations, patterns, and trends in data with temporal characteristics are available. In [20], authors presented a visualization of energy statistics using an existing web-based data analysis tools, including IBM's Many Eyes, and Google Motion Charts. In [15], authors presented a Java-based infrastructure, named SOCR Motion Charts, designed for exploratory analysis of multivariate data. SOCR is developed as a Java applet using object-oriented programming language. The authors successfully validated this visualization paradigm using several publicly available datasets containing housing prices or consumer price index.

A pair of online assessments designed to measure students' computational thinking skills were presented in [21]. The assessments represent a part of a larger project that brings computational thinking into high school STEM classrooms. Each assessment included interactive tools that highlight the power of computation in the practice of the scientific and mathematical inquiry. The computational tools including Google Motion Charts used in the assessments enabled students to analyze data with dynamic visualizations and explore concepts with computational models.

Successful visualizations of language changes using the diachronic corpus data were presented in [22]. In two case studies, authors illustrated recent changes in American English. In the first study, they visualized changes in a diachronic analysis of nouns and verbs. In the second study, they showed structural changes in the behavior of complement-taking predicates. They emphasized that MC are useful for the analysis of multivariate data over time and concluded that viewing the resulting data points in separate time slices offers a proper representation of the complex linguistic changes.

In [23], authors incorporated examples using recent business and economic data series and illustrated how MC can tell dynamic stories. They utilized a database of Bureau of Labor Statistics which publishes data on inflation, prices, employment, and many other labor related subjects. For the first analysis, they utilized the data about Current Employment Statistics and presented differences between the perception of common static tables and graphs, and the

dynamic nature of MC. They concluded that the static presentation style serves well the purpose of relaying accurate and non-biased quantitative data to analysts. Subsequently, they utilized the same data, but imported them to Google Docs. By loading the Motion Charts Gadget within the spreadsheet, they generated MC and visualized several areas of Labor Statistics. They emphasized that the benefit of MC lays in displaying complex multidimensional data changing over time on a single plane with the dynamic and interactive features. Users are then allowed to easily explore, interpret, and analyze the information in the data. They concluded that MC is an excellent and interesting way how to present valuable information that may be otherwise lost in the data.

The report on the implementation of AA in a new medical school can be found in [24]. Authors pointed out that analytics address two challenges in the curriculum: providing the evidence of the appropriate curriculum coverage and assessing the student engagement during the clinical placement. The paper describes tools and approaches applied on the data gained from their web-based clinical log system. The authors utilized common data visualization methods and examined their potentials to generate important questions. They also examined the value of a flexible approach to select the tools, the need for relevant skills, and the importance of keeping the viewer's attention. Subsequently, they utilized more sophisticated visualization methods, namely MC and Tree map. Using MC, they mapped several important variables including entry date, frequency of entries, clinical problems, the level of involvement, and the level of confidence. The authors appreciated the benefits of comparison of the variation of the frequency of entries, the confidence, and the level of involvement between students. The authors concluded that AA analysis using visualizations have already been a critical enabler of educational excellence, but there is undoubtedly further potential.

A beneficial feature for better visual perception of changes in time-series analysis is presented in [25]. Initially, the author highlighted the need for effective ways to examine quantitative data that changes over time and also noted that according to several studies, more than 70 percent of all business charts display time-series information. Then, the author emphasized both the benefits and the drawbacks of common data visualization methods, namely line plots and bar charts. Subsequently, the author described issues with the time-series analysis and presented capabilities of MC. The author pointed out that patterns of changes over time can take many meaningful forms and introduced a new feature, called visual trails, specially designed for MC. The feature allows seeing the full path for each variable from one point in time to another. It can be used for overcoming visual perception limitations of MC and allows analysts to examine degree of change, shape, velocity, and direction of change. Finally, the author conducted the experiment as an evaluation of the proposed improvement.

4. THE EDAIME TOOL

The preliminary version of the EDAIME tool was presented in [26]. We also described the results originated from the analysis of AA data. We utilized the data stored in the Information System of Masaryk University. The motivation to develop an enhanced version of MC was to improve its expression capabilities, as well as to facilitate analysts to depict each student or study as a central object of their interest. Moreover, the implementation enhances the number of animations that express the students' behavior during their studies more precisely. We validated usefulness of the developed methods with a case study where we successfully utilized the capabilities of the tool for the purpose of confirming our hypothesis concerning student retention. Although, we concluded that the methods proved to be useful for analytic purposes, more adjustments are needed.

Two main challenges are addressed by the presented VA tool. EDAIME enables visualization of multivariate data and the qualitative exploration of data with temporal characteristics. The technical advantages over other implementations of MC are its flexibility and the ability to manage many animations simultaneously. The Force Layout component of D3[1] provides the most of the functionality behind the animations and collision detection utilized in the interactive visualization methods. Technical aspects of enhanced MC methods are elaborately described in [27]. Investigated data can be imported directly using the tool. In cases where datasets have missing values at the beginning or the end, the missing values are extrapolated from nearby data. In other cases, gaps are filled with interpolated values. For the purposes of the MC analysis, it is not important that the data are not entirely accurate.

In two figures below, two examples of our enhanced MC methods can be seen. We already utilized the methods to verify a hypothesis concerning student retention. Figure 1 depicts a snapshot of the method captured in the second semester. Each element represents a field of study and consists of a pie chart. It allows analysts to investigate another data dimension easily. Each pie chart animates a relationship between finished and unfinished studies where the green sector quantifies the complete ones, and the red sector quantifies the others. Figure 2 represents a snapshot of the second method utilized for the same dataset also captured in the second semester. The large clusters of elements represent the particular field of study consisting of small elements that represent individual students. Therefore, the size of the cluster of elements corresponds to the number of students enrolled in the particular field of study. The size of the small elements determines the number of credits gained by students in the particular semester of the study. Besides the study progress, the animations are also utilized to express the study termination, the change of the mode of study and the change of the field of study. During the animation process, dropout students turn red and fall down the chart in the semester when they left the study. The stroke-width of the elements represents states of the study and the element color represents attributes of the study.

When animations are used for exploratory analysis of unfamiliar data, analysts do not know what elements are important and play the animation hoping that something emerges. Analysts may determine areas that look promising and replay the animation several times focusing on each of the potentially interesting areas in depth. This can become an issue, perhaps making trend animations slower and more error prone for analyses. If there is a lot of variability in the data, there will be a lot of random motions, making hard to perceive trends. If there are too many elements, a clutter and counter-trends can easily intricate an observation. In the next section, we describe several user interface features that may solve some of these issues. Naturally, all methods using animations have several limitations, but appropriately designed user interface features can considerably aid visual inspection of data.

[1] http://d3js.org/

4.1 User Interface Features

The EDAIME tool offers several beneficial configurable interactive features for a more convenient analytic process. User interface features are highly customizable and allow analysts to arrange the display and variable mapping according to his or her needs. Available features include a mouse-over data display, color and plot size representation, traces, animated time plot, variable animation speed, changing of axis series, changing of axis scaling, distortion, and the support of statistical methods.

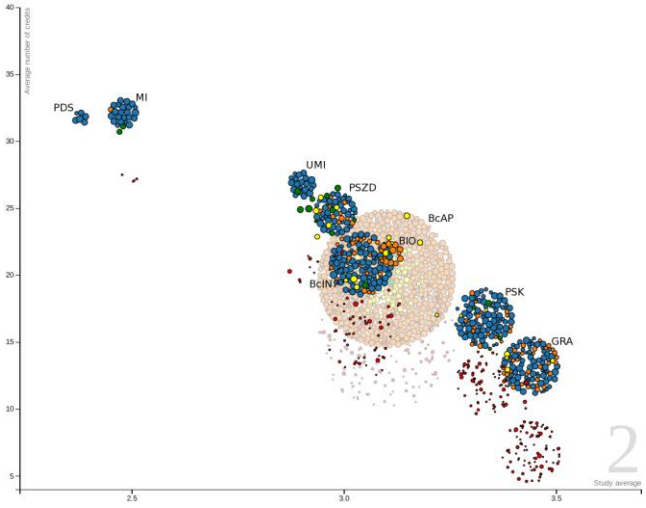


Figure 1. EDAIME snapshot: clusters of students.

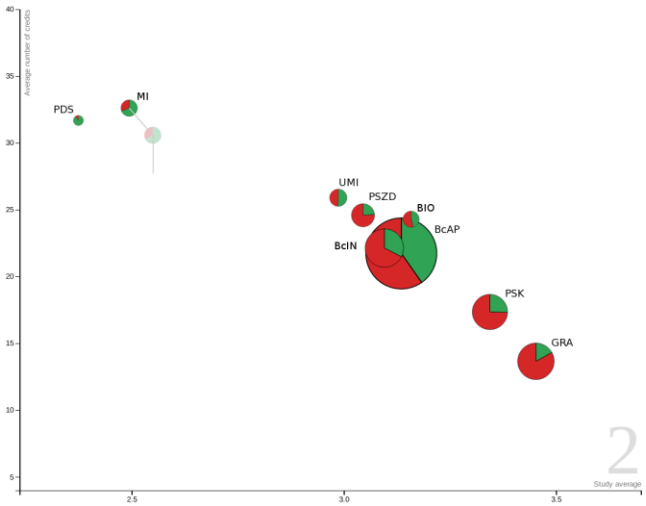


Figure 2. EDAIME snapshot: additional dimension using pie charts.

The focus-plus-context technique allows to interactively exploring objects of interest in detail while preserving the surrounding context. More precisely, if an analyst zooms in for detail, the chart area is too big to full overview. Contrarily, if an analyst zooms out the screen to see the overall chart area, the tiny but potentially important characteristics can disappear. Generally, distortions are particularly beneficial to overcome the aforementioned issues. The circular distortion magnifies the area around the mouse pointer, while leaving the chart area unaffected for the context. This distortion is useful especially to distinguish individual elements in a cluster. However, the area near the circumference of the elements is then compressed. Therefore, it is not suitable for representing quantitative values. However, a function which magnifies the details continuously in order to avoid such local errors exists. It applies the distortion to each dimension separately which results in Cartesian distortion. If elements overlap each other during the animation, it will be more difficult to track their paths. Using the jitter feature, a better visual perception of data can be obtained by adding small random quantities to all elements' values before displaying them. As mentioned earlier, it is not important that the data are not entirely accurate for the purposes of a trend analysis.

Regardless of the power of a human brain, a memory is limited. It is difficult to reconstruct the past events from a memory, to recapture the sequence of events and details of each moment. The tool provides analysts with the ability to select particular elements and show a trace for each of the selected elements as it progresses. This is particularly useful in verifying apparent anomalies noticed during an animation. The traces show elements at each location and sizes for each time point. The traces are then connected with edges to help clarify their sequences. Analysts can observe any interesting element while the previous states are still fresh in their memory. Anomalies emerge and can be examined even without animations, so analyses may be faster and less error prone. Points that move continuously through a range of values appear as clear trends. One key challenge must be addressed in the design of this view. The trend line direction must be made visually expressive, because there is no animation to indicate the direction. We solved this problem by using element transparency, fading from mostly transparent in the earliest elements to mostly opaque in the latest elements in the sequence. In order to perceive the flow direction even for smaller elements we employed the same approach with lines connecting the elements. In addition, it was necessary to render larger bubbles first to avoid occluding smaller bubbles. As described in [25], traces are particularly useful to reveal the nature of change and can help to examine the magnitude, shape, velocity, and direction of changes.

The support of statistical methods is also useful for examining the nature of change. The statistics provide simple summaries that form the basis of the initial description of the data and also serve as a part of a more extensive analysis. We implemented several measures that are commonly used to describe a dataset, i.e. measures of central tendency or measures of variability. The measures may be beneficial when identifying meaningful data characteristics of changes over time. We utilized both the univariate and the bivariate statistical methods. Input parameters for statistical methods consist of investigated MC variables. When an animation is running, each statistical measure is computed for every element on the background. Any combination of measure and variable can be selected using the user interface. The list of univariate measures includes coefficient of variation, skewness, mean, variance, standard deviation, median absolute deviation, median, geometric mean, and interquartile range. The mouse-click event on any element will extract an interactive HTML table on the right side of the chart area. The table consists of the measure computed for every element sorted in the descending order of the specified variable. If analysts select a row, the corresponding element will be highlighted. More precisely, the other elements are either transparent or hidden. Bivariate measures can be applied to any pair of variables. The list of bivariate measures includes sample covariance, sample correlation, and paired t-test.

The layout of the EDAIME user interface is presented in Figure 3. Using control, analysts can pause and advance the animation or change the speed. The Play, Pause, and Restart buttons are situated in the upper right corner next to the chart area. Above the buttons, the time slider is situated. Analysts can grab the time slider control to adjust the playback speed. Traces control is situated beneath the control buttons and it allows selecting elements of the interest to show their traces. This makes the selected elements more distinguishable and solves clutter issues.

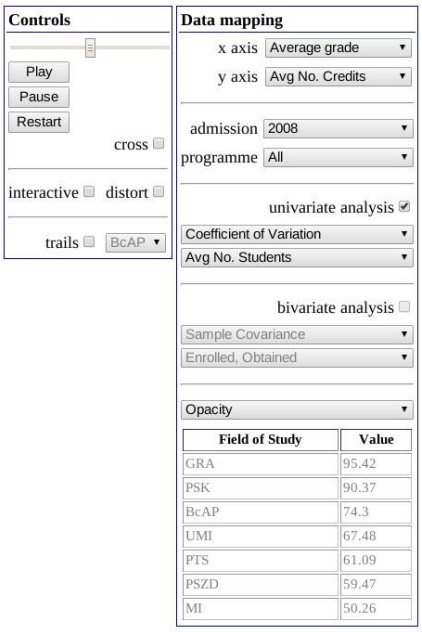


Figure 3. The EDAIME user interface layout.

5. EXPERIMENTATION

Any quantitative research of AA also requires a preliminary exploratory data analysis. Though useful, MC involves several drawbacks in comparison with common data visualization methods. Thus, empirical data is needed to evaluate its actual usability and efficacy.

In this section, we describe the experiment for the purpose of evaluating the efficacy of the enhanced MC methods implemented in EDAIME. We present the results including a detailed discussion. Twenty-two subjects (9 females, 13 males) with the average age of 31.6 (SD = 6.8) participated in our experiment. The participants ranged from 24 to 46 years of age. All participants came from professions requiring the use of data visualizations, including college students, analysts, and administrators. The experiment was conducted using standard desktop PCs. All subjects performed the experiment on an Intel Core i3 PC with 4 GB of RAM running Windows 7 or Fedora Core 20. Each PC had a 24" LCD screen running at the resolution of 1920 x 1080. We prefer Chrome as a web browser as it excellently supports HTML5 and CSS3 standards.

We performed a study to validate the usefulness and the general applicability of the enhanced version of MC in comparison with common data visualization methods when employed to analyze study related data. The experiment used a 4 (visualization) x 2 (size) within-subjects design. The visualizations varied between the static and the animated methods. The static methods were represented by line plots (LP) and scatter plots (SP) which were generated for each semester. The animated methods were represented by the standard MC with the basic user interface (BMC) and the enhanced MC with advanced user interface features (EMC) described in the previous section. The size of datasets varied between small and large ones with the threshold of 500 elements. For the experiment, we utilized study related data about students admitted to bachelor studies of the Faculty of Informatics Masaryk University between the years of 2006 and 2012.

5.1 Hypotheses and Tasks

We designed the experiment to address the following three hypotheses:

- H1. BMC methods will be less effective than static methods when used for small datasets, and more effective when used for large datasets. In other words, the participants will be (a) faster and (b) make fewer errors when analyzing large datasets using BMC methods.
- H2. EMC will be more effective than the other methods for all datasets. In other words, the participants will be (a) faster and (b) make fewer errors when using EMC methods for all dataset sizes.
- H3. The participants will be more effective with small datasets than with large datasets. In other words, the participants will be (a) faster and (b) make fewer errors when analyzing small datasets.

In each trial, the participants completed 16 tasks, each with 1 to 5 required answers. Each task had students' IDs as the answer. Several questions have more correct answers than requested. The participants were asked to proceed as quickly and accurately as possible. In order to reduce learning effects, the participants were told to make use of as many practice trials as they needed. We also instructed them to practice until they had reached the desired performance level. Moreover, the participants had access to the tool several days before the experiment.

Sample of tasks:

- Select 4 students whose rate of enrolled credits was faster than their rate of obtained credits.
- Which student had the most significant decrease of the average grade?
- Select 5 students with the significant increase of the number of credits.
- Select 3 students whose average grade increased first and decreased later.
- Which student had the most significant increase in the number enrolled credits?

The participants selected answers by selecting student IDs in legend box located in the upper right from the chart area. In order to complete the task, two buttons can be used–either "OK" button to confirm the participant's choice or "Skip Question" button to proceed to the next task without saving the answer. There was no time limit during the experiment. For each task, the order of the datasets was fixed with the smaller ones first. This also allowed the participants to build their skills as they proceeded.

5.2 Study Method

The experiment used a 4 (visualization) x 2 (size) within-subjects design. Each experiment block was preceded with a training session in which we showed the subjects the correct answers after they confirmed it to allow participants to get familiarized with the settings and UI. It was followed by 16 tasks (8 small dataset tasks and 8 large dataset tasks in this order). After that, the subjects completed survey with questions specific for the visualization. Each block lasted about 2 hours. The subjects were screened to ensure that they were not color-blind and understood common data visualization methods. We also attempted to balance gender. The study results are divided into three sections: accuracy, completion time, and subjective preferences. To test for significant effects, we conducted repeated measures analysis of variance (ANOVA). Only significant results are reported. Post-hoc analyses were performed by using the Bonferroni technique.

5.3 Accuracy

Since some of the tasks required multiple answers, accuracy was calculated as a percentage of the correct answers. Thus, when a subject selected only three correct answers from five, we calculated the answer as 60 % accurate rather than an incorrect answer. The analysis revealed several significant accuracy results at the .05 level. The type of visualization had a statistically significant effect on the accuracy for large datasets ($F(1.930, 40.535) = 25.655$, $p < 0.001$). Figure 4 illustrates graph of the mean accuracy of visualizations for large datasets including error bars that show the 95% confidence interval. Pair-wise comparison of the visualizations found significant differences showing that both animated methods were significantly more accurate than the static methods. EMC was more accurate than LP ($p = 0.001$). EMC was also more accurate than the BMC ($p < 0.001$). LP were more accurate than SP ($p = 0.016$). For small datasets, visualizations were not statistically distinguishable, except for SP which had lower accuracy than other methods. Also, the subjects were more accurate with small datasets ($F(1, 21) = 38.679$, $p < 0.001$) as can be seen in Figure 5.

5.4 Task Completion Time

An answer was considered to be incorrect if none of the correct answers was provided. In terms of time to task completion, we also observed a statistically significant effect ($F(1.764, 37.044) = 43.875$, $p < 0.001$). Post-hoc tests revealed that BMC was the slowest for both dataset sizes. For large datasets, the LP was faster than the EMC ($p < 0.001$). EMC and SP were not statistically distinguishable. The mean time for LP was 76.36 seconds compared to 85.95 seconds for the EMC–about 13% slower, 88.59 seconds for the SP–about 16% slower, and 91.64 seconds for the BMC–about 20% slower. For small datasets, static methods were significantly faster than animated. Pair-wise comparison of the visualizations found significant differences between all of them except for EMC and SP. LP were the fastest for all datasets. EMC was slower than the LP ($p < 0.001$) and faster than the BMC ($p < 0.017$). The mean time for BMC was 70.18 seconds compared to 67.6 seconds for the SP–about 3% faster, 66.55 seconds for the EMC–about 6% faster, and 61.36 seconds for the LP–about 14% faster.

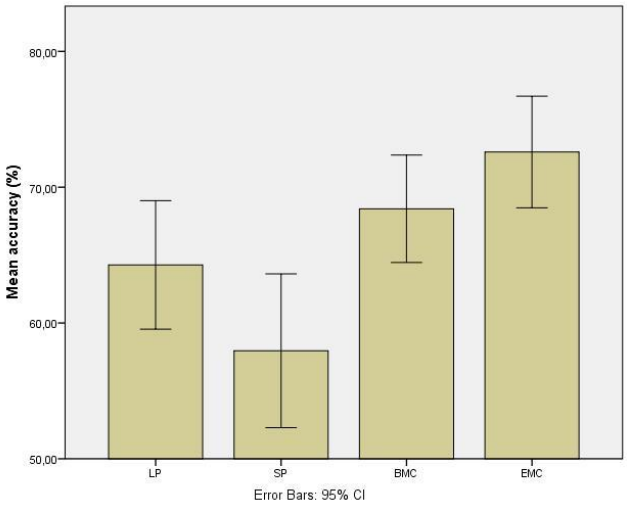


Figure 4. Mean accuracy of answers per visualization method.

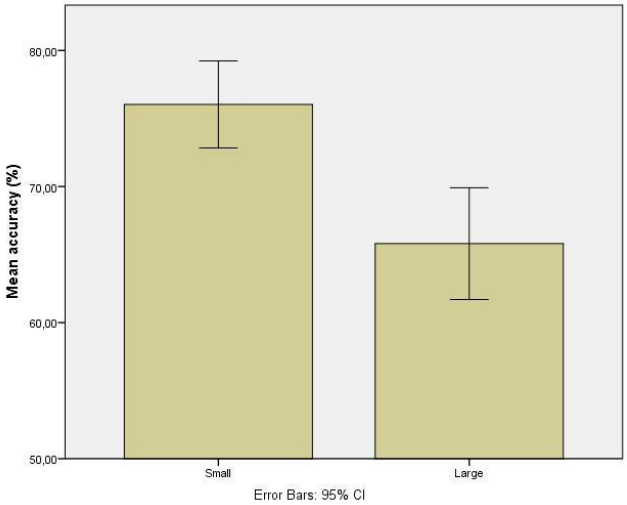


Figure 5. Mean accuracy of answers per dataset size.

5.5 Subjective Preferences

For each experiment block, the subjects completed a survey where the subjects assessed their preferences regarding analyses. The subjects rated the static and animated methods on a ten-point Likert scale (1 = strongly disagree, 10 = strongly agree). Using RM-ANOVA, we revealed statistically significant effects ($F(1.696, 35.611) = 80.1332$, $p < 0.001$). Post-hoc analysis found that EMC was significantly more helpful than other methods, more precisely BMC ($p < 0.001$) and LP ($p< 0.001$). The obtained results are presented in Table 1, indicating the resulted mean values of the preferences for each question.

The significant differences indicate that animated methods were judged to be more helpful than the static methods. The subjects significantly preferred the LP to use for small datasets. However, animated methods were judged to be more beneficial than static methods for large datasets ($p < 0.001$). The results also showed that

animated methods were more entertaining and interesting than the static methods ($p < 0.001$).

Table 1. The resulted mean values of the preferences.

	LP	SP	BMC	EMC
The visualization was helpful in answering the questions.	5.41	4.27	6.86	7.55
I found this visualization entertaining and interesting.	5.36	5.14	7.14	8.05
I prefer visualization for small datasets.	6.70	4.41	5.59	5.82
I prefer visualization for large datasets.	5.90	5.18	7.41	8.32

6. DISCUSSION

Our first hypothesis (H1) was that BMC would outperform both the static methods for large datasets and will be less effective when used for small dataset. This hypothesis was confirmed only partially. BMC methods were more accurate than the static methods, but contrary to the hypothesis, the static methods proved to achieve better speed than the BMC for the both dataset sizes. Moreover, the methods were not statistically distinguishable in terms of accuracy for small datasets. The second hypothesis (H2) expected that EMC will be more effective than the other methods for all dataset sizes. The hypothesis was only partially confirmed as well. EMC was the most accurate method for all datasets. Contrary to the hypothesis, LP was the fastest method for all datasets. We also hypothesized that the accuracy will be higher for smaller datasets (H3). The hypothesis H3.a was supported, because the subjects were faster with small datasets. The mean time for large datasets was 85.64 seconds and for small datasets was 66.42 seconds. The hypothesis H3.b was also supported, because the subjects committed fewer errors with small datasets when compared with large datasets. Generally, the accuracy is the issue for static visualizations when large datasets were employed.

The EDAIME tool facilitates users to utilize the enhanced MC methods with advanced interactive features. After the experiment, multiple subjects reported that they make use of advanced user interface features and spent a lot of time exploring the data during the practice trials. In the final discussion, the several subjects reported that the animations were entertaining and interesting. Contrarily, several subjects reported that for large datasets as the number of elements rose they experienced increasing difficulty to identify and remember the element of their interest that they were following and without user interface features it would be hard to handle it. The overall accuracy was quite low in the study with average about 70%. However, only three questions were skipped.

The study supports the intuition that using animations in analysis requires convenient interactive tools to support effective use. The study suggests that EMC leads to fewer errors. Also, the subjects found MC methods to be more entertaining and exciting. They slightly preferred it to the static method. The evidence from the study indicates that the animations were more effective at building the subjects' comprehension of large datasets. However, the simplicity of static methods was more effective for small datasets. These observations are consistent with the verbal reports in which the subjects refused to abandon the static visual methods generally. This finding illustrates that interest in animations does not preclude the subjects' appreciation of common methods. Overall, the participants would prefer to utilize both types of visual methods. Results supported the thoughts that MC does not represent a replacement of common statistic data visualizations but a powerful addition.

7. CONCLUSION AND FUTURE WORK

Commonly used static methods have principal limitations in terms of the volume and the complexity of the processed data. Animations are substantially transparent techniques that can present a good overview of the complex and large data. MC presents multiple elements and dimensions of the data on a single two-dimensional plane. The main contribution lies in enabling critical questions about data relationships and characteristics.

In the EDAIME tool, we enhanced the MC concept and expanded it to be more suitable for AA analyses. We also developed an intuitive, yet powerful, user interface that provides analysts with instantaneous control of MC properties and data configuration, along with several customization options to increase the efficacy of the exploration process. The tool provides a smart, convenient, and visually appealing way to identify potential correlations between different variables. We validate the usefulness and the general applicability of the designed tool with the experiment to assess the efficacy of the described methods in comparison with visual static methods.

The study suggests that animated methods lead to fewer errors for the large datasets. Also, the subjects find MC to be more entertaining and interesting. The entertainment value probably contributes to the efficacy of the animation, because it serves to hold the subjects' attention. This fact can be useful for the purpose of designing methods in learning settings. The more entertaining a method is, the easier it is to concentrate on the process and the more information can be acquired. The study also indicates that we need to appropriately adjust analytic tools when we begin to process time-varying, high-dimensional data. Especially, we need to focus on user interface features.

The current limitations of the tool are predominantly originated in the use of HTML5 standard, because there are still serious performance problems in several web browsers. Thus, only a certain number (generally less than 1000) of data points may be effectively visualized using animations. Features enabling effective data manipulation are essential. The additional representation of the data using enhanced MC methods gives analysts more possibilities in exploring the data.

We plan to create the synergy of EDAIME animated methods with common DM methods to follow the VA principle more precisely. We already implemented a standalone EDAIME method utilizing decision tree algorithm providing visual representation. We prefer decision trees because of their clarity and simplicity to comprehend.

8. ACKNOWLEDGMENTS

We thank all colleagues of IS MU development team and Knowledge Discovery Lab for their assistance. This work has been partially supported by Faculty of Informatics, Masaryk University.

9. REFERENCES

[1] Goldstein, P. J. 2005. Academic analytics: The uses of management information and technology in higher education. EDUCAUSE. Retrieved from https://net.educause.edu/ir/library/pdf/ers0508/rs/ers0508w.pdf.

[2] Campbell, J. P., DeBlois, P. B., and Oblinger, D. G. 2007. Academic analytics: A new tool for a new era. EDUCAUSE Review, 40-57.

[3] Romero, C. and Ventura, S. 2013. Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12-27. DOI: http://dx.doi.org/10.1002/widm.1075.

[4] Delavari, N., Phon-Amnuaisuk, S., and Beikzadeh, M. R. 2008. Data Mining Application in Higher Learning Institutions. Informatics in Education, 31-54.

[5] O'Reilly, T. and Battelle, J. 2009. Web squared: Web 2.0 five years on. DOI: http://dx.doi.org/10.4304/jait.2.4.204-216.

[6] Tversky, B., Morrison, J. B., and Betrancourt, M. 2002. Animation: Can It Facilitate?. International Journal Human-Computer Studies, 247-262. DOI: http://dx.doi.org/10.1006/ijhc.2002.1017.

[7] Margaret, S., Chan, J., and Black, B. 2005. When can animation improve learning?. Some implications on human computer interaction and learning. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, 933-938.

[8] Le, D.-T. 2013. Bringing Data to Life into an Introductory Statistics Course with Gapminder. Teaching Statistics, 114–122. DOI: http://dx.doi.org/10.1111/test.12015.

[9] Heer, J. and Robertson, G. 2007. Animated Transitions in Statistical Data Graphics. IEEE Transactions on Visualization and Computer Graphics, 1240-1247. DOI: http://dx.doi.org/10.1109/TVCG.2007.70539.

[10] Kehoe, C., Stasko, J., and Taylor, A. 2001. Rethinking the Evaluation of Algorithm Animations as Learning Aids: An Observational Study. International Journal of Human-Computer Studies, 265-284. DOI: http://dx.doi.org/10.1006/ijhc.2000.0409.

[11] Bertamini, M. and Proffitt, D. R. 2000. Hierarchical motion organization in random dot configurations. Journal of Experimental Psychology: Human Perception and Performance, 1371-86.

[12] Robertson, G., Cameron, K., Czerwinski, M., and Robbins, D. 2002. Animated Visualization of Multiple Intersecting Hierarchies. Information Visualization, 50-65.

[13] Few, S. 2009. Now You See It: Simple Visualization Techniques for Quantitative Analysis. Analytics Press. DOI: http://dx.doi.org/10.1080/10543401003641225.

[14] Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. 2006. Phosphor: explaining transitions in the user interface using afterglow effects. In Proceedings of the 19th annual ACM symposium on User interface software and technology. DOI: http://dx.doi.org/10.1145/1166253.1166280.

[15] Al-Aziz, J., Christou, N., and Dinov, I. D. 2010. SOCR Motion Charts: an efficient, open-source, interactive and dynamic applet for visualizing longitudinal multivariate data. Journal of Statistics Education, 18(3), 1-29.

[16] Grossenbacher, A. 2008. The globalisation of statistical content Statistical Journal of the IAOS. Journal of the International Association for Official Statistics, 133-144.

[17] Baldwin, J. and Damian, D. 2013. Tool usage within a globally distributed software development course and implications for teaching. Collaborative Teaching of Globally Distributed Software Development, 15-19. DOI: 10.1109/CTGDSD.2013.6635240.

[18] Sultan, T., Khedr, A., Nasr, M., and Abdou, R. 2013. A Proposed Integrated Approach for BI and GIS in Health Sector to Support Decision Makers. Editorial Preface.

[19] Yoon, S., Elhadad, N., and Bakken, S. 2013. A Practical Approach for Content Mining of Tweets. American journal of preventive medicine, 122-129. DOI: http://dx.doi.org/10.1016/j.amepre.2013.02.025.

[20] Vermylen, J. 2008. Visualizing Energy Data Using Web-Based Applications. American Geophysical Union.

[21] Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., and Wilensky, U. 2014. Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms. Intelligent Technologies for Interactive Entertainment, 22-25. DOI: http://dx.doi.org/10.1007/978-3-319-08189-2_3.

[22] Hilpert, M. 2011. Dynamic visualizations of language change: Motion Charts on the basis of bivariate and multivariate data from diachronic corpora. International Journal of Corpus Linguistics, 435-461. DOI: http://dx.doi.org/10.1075/ijcl.16.4.01hil.

[23] Battista, V. and Cheng, E. 2011. Motion Charts: Telling Stories with Statistics. JSM Proceedings, Statistical Computing Section, 4473-4483.

[24] Olmos, M. and Corrin, L. 2012. Academic analytics in a medical curriculum: enabling educational excellence. Australasian Journal of Educational Technology, 1-15.

[25] Few, S. 2007. Visualizing Change: An Innovation in Time-Series Analysis. In Visual Business Intelligence Newsletter, White paper SAS.

[26] Geryk, J. and Popelinsky, L. 2014. Analysis of Student Retention and Drop-out using Visual Analytics. In Proceedings of the 7th International Conference on Educational Data Mining. International Educational Data Mining Society, 331-332.

[27] Geryk, J. and Popelinsky, L. 2014. Visual Analytics for Increasing Efficiency of Higher Education Institutions. In Proceedings of the 6th Workshop on Applications of Knowledge-Based Technologies in Business, 117-127. DOI: http://dx.doi.org/10.1007/978-3-319-11460-6.

Data-driven Proficiency Profiling

Behrooz Mostafavi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
bzmostaf@ncsu.edu

Zhongxiu Liu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
zliu24@ncsu.edu

Tiffany Barnes
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
tmbarnes@ncsu.edu

ABSTRACT

Deep Thought is a logic tutor where students practice constructing deductive logic proofs. Within Deep Thought is a data-driven mastery learning system (DDML), which calculates student proficiency based on rule scores weighted by expert-decided weights in order to assign problem sets of appropriate difficulty. In this study, we designed and tested a data-driven proficiency profiler (DDPP) method in order to calculate student proficiency without expert involvement. The DDPP determines student proficiency by comparing relevant student rule scores to previous students who behaved similarly in the tutor and successfully completed it. This method was compared to the original DDML method, proficiency based on average rule scores, and proficiency based on minimum rule scores. Our testing has shown that while the DDPP has the potential to accurately calculate student proficiency, more data is required to improve it.

Keywords

Data-driven, Tutoring system, Student classification

1. INTRODUCTION

Data-driven methods, methods where each step and calculation is based on analyzing a set of historical data, have been used to great effect to improve individualized computer instruction. They have been used in intelligent tutoring systems to accurately predict student behavior and improve learning outcomes. In contrast to individualized tutoring systems based on developing complex and context specific models of behavior, data-driven systems reduce the need for expert involvement to design the system, and can potentially adapt to new users without refinement of a behavioral model. This is because data-driven systems analyze previous student data in order to model student behavior and determine the best course of outcome in the tutor. Therefore, developing a data-driven intelligent tutoring system is based on gathering data, and developing the methods the system uses to analyze and react to student behavior.

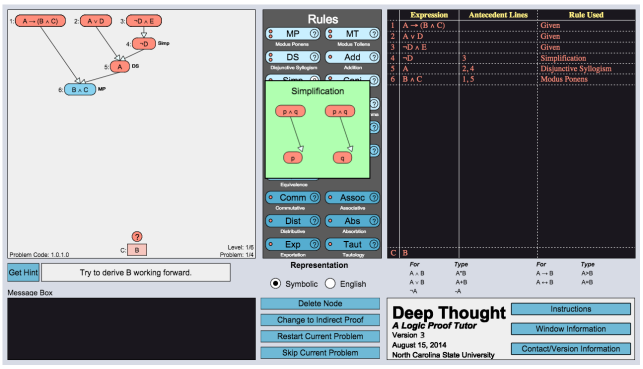


Figure 1: The Deep Thought DT3 logic tutor. Students apply logic rules (axioms) to premises to derive new statements until the conclusion (at the bottom) is justified. The right window displays the proof in standard list format.

We have been incrementally augmenting the Deep Thought logic tutor (Fig. 1) with data-driven methods for formative feedback and problem selection to improve student learning and reduce tutor dropout. Our long term goal is to create an intelligent tutor for logic proof construction that is fully data-driven and can adapt to students learning logic with varying curricular requirements without the need for further expert input. To this end, the next step in our work is to replace the expert-authored assessment parameters built into our problem selection system with a data-driven proficiency calculation that approximates the original system's performance.

Deep Thought utilizes a data-driven mastery learning system (DDML) consisting of 6 strictly ordered levels of proof problems. Each level is split into a higher proficiency track with a lower number of complex problems, and a lower proficiency track with a greater number of simpler problems. The first level of problems are the same for all students, and are used to estimate their initial proficiency. Proficiency is calculated using the knowledge tracing of all rule-application actions taken in the tutor. These action scores are compared to the average score thresholds of corresponding problems solved by past *exemplars* – students who have successfully completed the entire tutor, and have therefore demonstrated sufficient proficiency in the subject matter (Fig. 2).

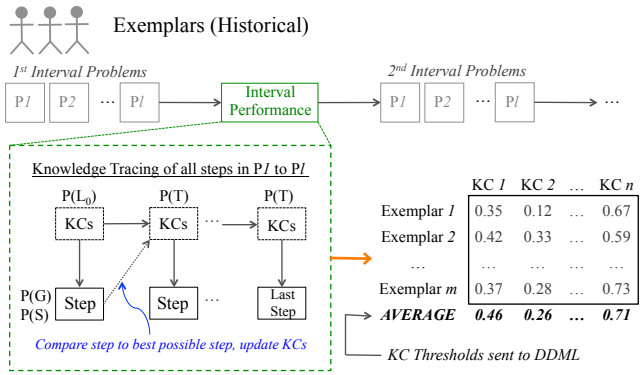


Figure 2: The DDML's threshold builder. Knowledge components (KCs) for each exemplar are updated using action steps from an interval set of tutor problems. The KC score averages at each interval are used as thresholds in the DDML system.

The difference between each action score minus its threshold is weighted by the expert-decided *priorities* of those actions within the level (Eq. 1). The sign of the resulting score determines placement in either the higher (+) or lower (−) proficiency track. On each subsequent level the system will first estimate a student's proficiency and then assign them to the higher or lower proficiency track based upon their prior performance. This system was shown to increase student completion and reduce tutor dropout over unordered and hint-based versions of Deep Thought [10].

Level l End Proficiency =

$$\textbf{sign}\left[\sum_{i=rule_0}^{rule_n}(scoreSign_{l,i} \times rulePriority_{l,i})\right] \quad (1)$$

Since the current DDML system uses expert-decided priorities for each of the rule application actions when calculating a student's proficiency, any new problems or levels added to the system will require expert involvement to determine which rules were prioritized in each new or altered level. This paper describes a study to develop a data-driven method of determining student proficiency that can replace the current expert-decided rule priorities in Deep Thought. This Data-driven Proficiency Profiler (DDPP) uses the clustering of exemplar scores at each level interval for each rule, weighted by primary component importance, to classify exemplars into *types* of student progress through the tutor. New students using the tutor will be assigned to a proficiency track based on comparison to existing types.

The DDPP method is compared alongside proficiency calculations using the minimum rule scores and average rule scores of exemplars, also weighted by primary component importance, to see how these methods compare to each other and to the expert authored system. We hypothesize that the DDPP will perform more accurately than the minimum or average methods of student proficiency classification. This would allow Deep Thought to be used in other classrooms where the pedagogical method and problem-solving ability of the class may be disparate from the current exemplar data from Deep Thought.

Our results show that proficiency calculation using average rule scores performs more accurately than proficiency calculated based on minimum rule scores. In addition, the DDPP method performs more accurately than the average method in some parts of the tutor, while it is less accurate in other parts. Unfortunately, the DDPP system does not yet reach the accuracy of the original system overall in calculating student proficiency. We conclude that more data is required in order for the DDPP to properly approximate the accuracy of the original system's proficiency calculation.

2. RELATED WORK

2.1 Data-driven Tutoring

An early example of a data-driven intelligent tutor is the Cognitive Algebra Tutor[12]. Here the authors introduce an algebra tutor which models student behavior based on the cognitive theory ACT-R and student data gathered from several previous studies. The Cognitive Algebra Tutor was several years and studies into development at this time, and the result is an example of a mostly-realized data-driven tutor. The tutor as it stood improved student performance, and the authors noted that although it over-predicted student performance, it would be improved the more data was collected. However, this system still took a long time and a great deal of expert involvement to design and improve. Conversely, developing a data-driven method of student assessment would reduce this time and effort, since it would be based on analyzing previous data rather than developing and improving on a cognitive model.

Later analyses on the potential benefits, and recommendations, for using data-driven methods to develop intelligent tutoring systems have focused on improving the modelling of student behavior rather than using data to improve on student assessment. Koedinger et al[7] give a very detailed overview on developing data-driven intelligent tutoring systems, and techniques for incorporating data in a useful way. They discuss optimizing the cognitive model using learning factors analysis; fitting statistical models to individual students; modeling student mood and engagement by modeling off-task behaviors, careless errors, and mood; and improving how the tutor selects actions for the student via MDP or POMDP. In a later work[8] the authors compare and contrast current data-driven methods for intelligent tutoring and discuss the potential for these methods to improve MOOCs. They go over the success cases for using data to improve tutors and coursework, in particular cognitive task analysis.

There have been several recent studies that demonstrate the potential for data driven methods to result in tutors that more accurately assess student performance and react to student behavior. Lee and Brunskill[9] examined the benefits and drawbacks to basing model parameters on existing data from individual students in comparison to data from an entire population, specifically as it pertained to the number of practice opportunities a student would require (estimated) to master a skill. The authors estimated that using individualized parameters would reduce the number of practice opportunities a student would need to master a skill. Gonzalez et al.[4] demonstrated a data-driven model which au-

tomatically generated a cognitive and learning model based on previous student data in order to discover what skills students learn at any given time, and when they use skills they have learned. The resulting model predicted student behavior without the aid of previous domain knowledge and performed comparably to a published model.

Data-driven intelligent tutors not only have the potential to more accurately predict student behavior, but interpret why it occurs. For instance, Elmadani et al. [2] proposed using data-driven techniques to detect student errors that occur due to genuine misunderstanding of the concepts (misconception detection). They processed their data using FP-Growth in order to build a set of frequent itemsets which represented the possible misconceptions students could make. The authors were able to detect several misconceptions based on the resulting itemsets of student actions. Fancsali[3] used data-driven methods to detect behaviors that usually detract from a student's experience with an ITS (off-task behavior, gaming the system, etc).

2.2 Cluster-based Classification

Cluster-based classification has several advantages when applied to data-driven tutoring. New educational technologies may reveal unexpected learning behaviors, which may not yet be incorporated in expert-decided classification processes. For example, Kizilec et al. [5] clustered MOOC learners into different engagement trajectories, and revealed several trajectories that are not acknowledged by MOOC designers. In addition, experts classify using their perception of the average students' performance[11] [13]. This perception may be different from the actual participant group. Cluster-based classification methods, however, are able to classify and update classifications based on actual student behaviors.

Moreover, previous studies have shown that personalized tutoring based on cluster-based classification not only helps learning, but improves users' experience. Klasnja-Milicevic et al. [6] gave students different recommendations on learning content based on their classified learning styles. As a result students who used hybrid recommendation features completed more learning sessions successfully, and perceived the tutor as more convenient. Despotovic-Zrakic et al. [1] adapted different course-levels, learning materials, and content in Moodle, an e-learning platform, for students in different clusters. Results showed that students with adapted course design had better learning gain, and a more positive attitude towards the course.

However, the majority of previous work clustered students solely on their overall performance statistics. In contrast, our method clusters students based on their application of specific knowledge components throughout the tutor.

3. METHODS

The Data-driven Proficiency Profiler (DDPP) is a system which calculates student proficiency at the end of each level in Deep Thought based on how a given student performs in comparison to exemplars who employed similar problem solving strategies (see Fig. 3), with rule scores weighted as determined through principal component analysis. Based on how similar exemplars were assigned in subsequent levels, the DDPP can determine the best proficiency level for a new student. In contrast to the DDML system previously employed, this proficiency calculation and rule weighting is entirely data-driven, with no expert involvement. We hypothesize that the DDPP based calculation will perform more accurately when compared to average and minimum methods.

3.1 Data-driven Problem Profiler

We first determined similar problem solving strategies among the exemplars by clustering the exemplars' rule scores (KCs) based on hierarchical clustering. For the initial single-point distance measure we used Euclidean squared distance, while for the hierarchical clustering algorithm we used cluster centroids to determine the distance between individual clusters. As a result each exemplar is assigned to a set of n clusters (where n is equal to the number of KCs), as shown in the table in Fig. 3.

Expert weighting was replaced by principal component analysis (PCA) of the frequency of the rules used for each exemplar for each level, accounting for 95% variance of the results. PCA is typically used to reduce the dimensionality of a data set by determining the most influential factors in the data set. The influence of a given factor is based on how much that factor contributes to the variability in the data. We use PCA analysis on the Deep Thought data set to determine which rules were most important to success in the tutor at each level. Rules which account for 25% of importance and higher are considered most important for completing a level. This percentage was determined through testing, and is the percentage that maximized accuracy. For each rule, its PCA importance value is the new weight for that rule score. Unlike expert authored weights, these rule score weights are based on each rule's importance as determined by the data.

When a new student uses the tutor, the student's rule scores are calculated throughout the level. At the end of each level, the DDPP looks at each student's individual rule score and assigns it to a cluster for that rule. The DDPP then finds which clusters the scores for the most important rules fall into for that level (based on the same PCA based weighting), and then classifies that student into a type based on the set of clusters the student matches (see Fig. 3, right). Finally the system assigns the student to a proficiency track based on data from the matching type of exemplars, and how those exemplars were placed in the next level. The more exemplars we have of a given type, the stronger the prediction we can make for a new student. In the event that a new student doesn't match an existing type in the exemplar data, their proficiency is calculated using the average scores. Average scores are used as a default because, as shown in the results, for most levels it is a better prediction approximation than using the minimum scores.

3.2 DDPP Advantages

In the original system, the student proficiency was determined based on one set of rule thresholds and a set of expert authored weights. However as a result, the system didn't take varying student problem-solving strategies into account. The data is based on students who completed the tutor, who have therefore shown the level of mastery required to successfully complete Deep Thought. However the

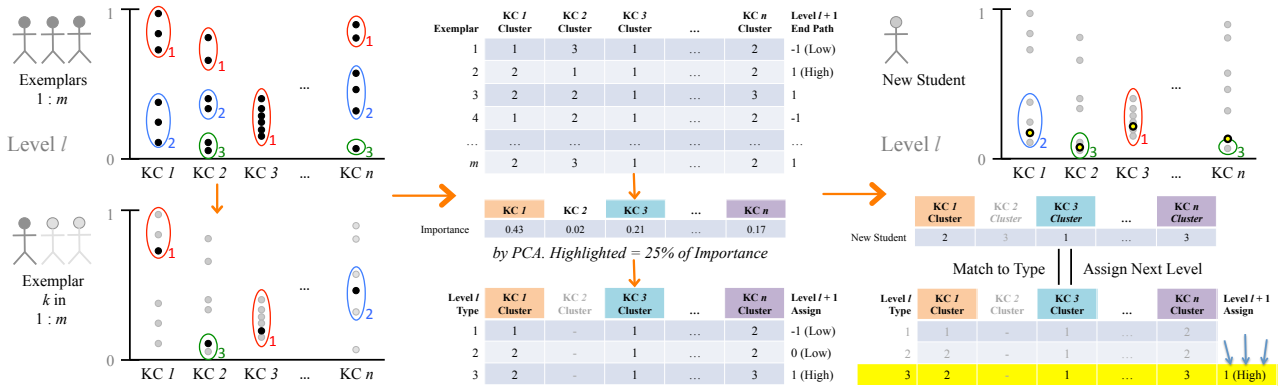


Figure 3: The Data-Driven Proficiency Profiler. (Left) At each level interval, exemplar KC scores are clustered, and exemplars are assigned a cluster for each KC Score. (Center) KCs that make up 25% of importance in the current level are used to assign exemplars to types. (Right) New student scores are assigned to clusters, and compared to existing types to determine next level path.

scores are averaged over all the students at the end of each level. By taking the average of these student scores at this point, we're still assuming only one successful problem solving strategy for completing each level in the tutor. However while most strategies might be the same for earlier levels, there may be a variety of strategies in later levels that can still result in successful completion.

The DDPP method accounts for that possible variety in problem solving methods. In using an unsupervised clustering method, we're able to account for different clusters while not knowing how many clusters there are for each rule. By clustering the scores, we're essentially looking for different strategies that utilize particular rules and determining these strategies based on the student data. Once we determine which strategy a new student is utilizing, we can look to the data again to see how exemplars who employed a similar strategy were placed in the tutor and how they performed, thus determining the best way for the tutor to react to that particular student. Using PCA based weights allows us to weight rule scores based on rule importance as determined by previous students who completed the tutor, rather than expert determination.

3.3 Evaluation

Testing was performed on data collected from two courses using Deep Thought with the DDML system. The first was a Philosophy deductive logic course ($n = 47$) using Deep Thought as a regular assignment over the course of a 15-week semester. The second was a Computer Science discrete mathematics course ($n = 84$), using Deep Thought as a two week assignment during the course's 4-week logic curriculum. From the students in these data, 26 of the Philosophy students (55%) and 50 of the Computer Science students (60%) completed the tutor, and were used as exemplars for the compared methods. By completing all levels in Deep Thought, these students have demonstrated sufficient mastery of the skills needed for introductory proof problem-solving.

By using data from both Computer Science and Philosophy based teaching methods for propositional logic, we expand the range of problem solving strategies analyzed and exemplar types determined. This allows us to test the tutor's performance across different classroom conditions, and determine whether the methods for proficiency path placement are effective for students in different disciplines that use different teaching methods.

The DDML system used the average of exemplar rule scores, weighted by expert-authored end of level rule priorities, to calculate student proficiency. In total there were 19 individual rule actions in Deep Thought on which students were evaluated. Based on the results of this calculation, the DDML system determines whether to send a student on the higher or lower proficiency path in the next level. The system also allowed for the possibility of students switching proficiency paths in situations where the student cannot complete the level on the path they were originally assigned. Because students can switch paths in the middle of a level, we can determine if they finish the current level on the same path they were assigned. If the student did not finish the level on the same proficiency path, it is an indication that the DDML system may have initially assigned the student to the wrong proficiency path. Therefore we can calculate the accuracy of the original system by determining how often students who completed the entire tutor changed proficiency paths throughout. Given $S_{sameTrack}$ as the number of students who finished a level on the same proficiency track, and S_{total} as the total number of students who completed the level, the path prediction accuracy for each level ($LevelAccuracy$) is calculated as follows:

$$LevelAccuracy = \frac{S_{sameTrack}}{S_{total}} \quad (2)$$

The $LevelAccuracy$ for each level is added together to determine the path prediction accuracy. This calculation tells us, for students who completed the entire tutor, how well the original system predicted the paths for them to continue on. This serves as a basis of comparison between the DDPP and the original DDML system.

3.3.1 Minimum & Average

The average rule scores are the set of average scores for each rule in each level. Minimum scores are the smallest scores in the exemplar data set for each rule in each level. This calculation is based on the assumption that if a student scores at least at this minimum for a given rule in that level, the student should be able to perform as well as an exemplar throughout the tutor. The difference between the current DDML system and average score or minimum score based proficiency calculation is that the DDML weighted scores with expert-decided rule priorities, while average or minimum weighted average or minimum scores with PCA-determined weights. Calculating proficiency based on average and minimum scores offers insight into how introducing PCA to students' performance baseline changes the prediction accuracy.

4. RESULTS

The prediction accuracy of the minimum, average, and DDPP methods were calculated for the 76 exemplars from the Philosophy and Computer Science data sets. Ten-fold cross validation was used to train and test the methods across the combined data. We focus on the results of the path prediction accuracy described in section 3.3 as a basis of comparison between the original system, the DDPP, proficiency based on average scores, and proficiency based on base minimum scores. These results are in tables 1, 2, and 3.

4.1 Path Prediction Accuracy

Table 1 shows the path prediction accuracy of the DDML system, the DDPP system, average score assessment, and minimum score assessment across all the students in the Philosophy and CS courses. The original system accuracy was very high, ranging from 75% at the end of level 3 to 88.2% at the end of level 1. The DDPP was somewhat accurate, ranging from 61.8% path prediction accuracy at the end of level 4 to 67.1% path prediction accuracy at the end of level 2. While these accuracies are not nearly as high as in the original system, they are very good considering that, unlike the original system, path prediction in the DDPP is entirely data-driven. It should also be noted that the DDPP was more consistent in its accuracy, only varying by at most 5% between levels (in comparison to the original DDML system, which ranged in accuracy by 9.3%).

Table 1: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for both Philosophy and CS students at the end of each level

	Original	DDPP	Average	Minimum
Lvl 1	88.2%	65.8%	65.8%	35.5%
Lvl 2	85.5%	67.1%	73.7%	18.4%
Lvl 3	75.0%	63.2%	60.5%	69.7%
Lvl 4	78.9%	61.8%	64.5%	40.8%
Lvl 5	78.9%	64.5%	59.2%	59.2%

Overall the original system predicted paths more accurately than the DDPP, average, or minimum methods across all levels. The minimum method was least accurate across all levels. In comparison to the average method, the DDPP was more accurate than the average method at the end of levels 3 and 5. The DDPP was equally as accurate as the average method at the end of level 1, and less accurate at the end of levels 2 and 4. However, some of the lower accuracy was likely due to the distribution of exemplars across the two courses. Recall that the CS students made up a higher proportion of the analyzed exemplars than the Philosophy students. Analyzing the path prediction accuracy by the individual course reveals more detail on the path prediction accuracy.

4.2 Philosophy & CS Accuracy

In the case of the Philosophy students, where proportionally fewer of the students were selected as exemplars, the DDPP system was more accurate than the original system on every set of levels except for the end of level 5 (see Table 2). In comparison to the average calculation method, the DDPP was only more accurate at the end of level 3. At the end of levels 1 and 5, the DDPP was as accurate as the average method, and at the end of levels 2 and 4 the DDPP was less accurate.

Table 2: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Philosophy students

	Original	DDPP	Average	Minimum
Lvl 1	76.9%	80.8%	80.8%	23.1%
Lvl 2	65.4%	69.2%	76.9%	19.2%
Lvl 3	50.0%	84.6%	80.8%	38.5%
Lvl 4	65.4%	69.2%	76.9%	30.8%
Lvl 5	53.8%	46.2%	46.2%	26.9%

In the CS course, where proportionally more of the students were selected as exemplars, not only was the original system far more accurate than it was for the entire set of students overall, but the DDPP path accuracy was much worse in some places. However, in comparison to the average method, the DDPP method was only less accurate in level 2. In all other levels the DDPP was either more accurate than the average method (levels 3 and 5) or equally as accurate (levels 1 and 4).

Table 3: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Computer Science students

	Original	DDPP	Average	Minimum
Lvl 1	94.0%	58.0%	58.0%	42.0%
Lvl 2	96.0%	66.0%	72.0%	18.0%
Lvl 3	88.0%	52.0%	50.0%	86.0%
Lvl 4	86.0%	58.0%	58.0%	46.0%
Lvl 5	92.0%	74.0%	66.0%	76.0%

4.3 Discussion

In the original DDML method, the weight of each rule was determined by domain experts. Our results show that when replacing the original weights by weights determined through principal component analysis in the average score method, the prediction accuracy increases for all levels in the philosophy class, but decreases for all levels in the computer

science class. This may because the experts were computer science students and teachers, who prioritized rules with the performance of computer science students in mind. When the real participants were philosophy students, Principal Component Analysis outperformed experts because it prioritized rule based on the performance of the real participants. It's possible that expert involvement may be constrained by the expert's background, whereas a data-driven approach is more flexible when adapting to the diversity of participants.

When comparing the path prediction accuracy of the original method to the DDPP, our result shows that the DDPP calculated student proficiency with more accuracy in the case of the Philosophy students, but less accuracy overall or in the case of the Computer Science students. It is likely that these results are a product of the limited, uncontrolled nature of the dataset. Only 76 exemplars were chosen overall, and of those exemplars a disproportionate number of them were selected from the computer science course. We noticed in the data that the students in the Computer Science course had KC weights that were vastly different than the expert weights. This means the students in the Computer Science course were showing some unorthodox problem solving strategies, particularly in the earlier levels. With enough data and more students with varying strategies, the DDPP could more accurately assign other students who employ different proof solving strategies. However for this limited dataset, it is possible that there were not enough students employing the same unorthodox strategies that a type could be determined.

Table 4: The average number of types found per level during training (exemplars), and the number of students typed during testing (new students). There were a total of 76 students in the data set.

Level	**1**	**2**	**3**	**4**	**5**
Avg. Types Found (Train)	14	13	21	17	26
# Types Matched (Test)	0	4	2	2	10

Table 4 shows the average number of types found in the training dataset, and the number of students matched to a type during testing. While there were several types found in the training step, far fewer students could be matched to a type in the testing step. This would explain the lower accuracy in the DDPP system, as well as why it performed similarly to the average method; it is likely that many of the students in the test set could not be classified into a type, which would result in the DDPP using the calculation based on average scores to determine student proficiency.

That said, the DDPP is still very accurate considering that, in all aspects of proficiency calculation, it is completely data-driven. Its accuracy when applied to the students in the Philosophy class in particular shows the potential for this system to be useful in different classroom conditions. The clustering step at each level produced between 14 and 26 possible types of exemplars to compare students to, compared to what would have been 76 individual students in the original system. This results in a system of proficiency calculation that, given more data, has the potential to calculate student proficiency just as accurately and more efficiently as the original.

5. CONCLUSIONS & FUTURE WORK

We have presented a fully data-driven student proficiency calculator, the Data-driven Proficiency Profiler (DDPP). The DDPP clusters exemplar student data into types, attempts to classify new students into one of the exemplar types, and calculate proficiency based on exemplars who employed similar problem strategies. We hypothesized that the DDPP would be more accurate than proficiency calculated using average scores or minimum scores. Instead, our results showed that the DDPP performed about as well as the average method overall, and did not approximate the accuracy of the original system. However our data set was very limited, and the high accuracy the DDPP achieved for the Philosophy students shows this system has potential once more data can be acquired.

In the future, we would like to be able to test this system with more data. The more students use the system, the greater the data set we will be able to use and the more conclusions we will be able to draw on the qualities of the DDPP system. In particular we will analyze in greater detail the types found on each level and the differences between each type in terms of problem solving strategy. We can also determine the importance, in depth, of certain rules to each level and the problems within it based on student problem solving strategies. Our final step is to implement the DDPP into Deep Thought and use it to direct students through the levels. Implementing the DDPP into Deep Thought will allow us to test whether, ultimately, the DDPP is an accurate, data-driven proficiency calculation.

6. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants 1432156 and 0845997.

7. REFERENCES

[1] M. Despotovic-Zrakic, A. Markovic, Z. Bogdanovic, D. Barac, and S. Krco. Providing adaptivity in moodle lms courses. *Educational Technology Society*, 15(1):326–338, 2012.

[2] M. Elmadani, M. Mathews, and A. Mitrovic. Data-driven misconception discovery in constraint-based intelligent tutoring systems. In *Proceedings of the 20th International Conference on Computers in Education (ICCE)*, pages 26–20, 2012.

[3] S. E. Fancsali. Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, pages 28–35, 2014.

[4] J. P. Gonzalez-Brenes and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, pages 236–239, 2013.

[5] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In

Proceedings of the 3rd international conference on learning analytics and knowledge, pages 170–179, 2013.

[6] A. Klasnja-Milicevic, B. Vesin, M. Ivanovic, and Z. Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers Education*, 56(3):885–899, 2011.

[7] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[8] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Data-driven Learner Modeling to Understand and Improve Online Learning: MOOCs and technology to advance learning and learning research (Ubiquity symposium). In *Ubiquity 2014*. 2014.

[9] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, pages 118–125, 2012.

[10] B. Mostafavi, M. Eagle, and T. Barnes. Towards Data-driven Mastery Learning. In *To appear in Proc. Learning, Analytics, and Knowledge (LAK 2015)*.

[11] E. V. Perez, L. M. R. Santos, M. J. V. Perez, J. P. de Castro Fernandez, and R. G. Martin. Automatic classification of question difficulty level: Teachers' estimation vs. students' perception. In *Proceedings of the IEEE Frontiers in Education Conference*, pages 1–5, 2012.

[12] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic BulletinReview*, 14(2):249–255, 2007.

[13] G. van de Watering and J. van der Rijt. Teachers and students perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. *Educational Research Review*, 1(2):133–147, 2006.

Interaction Network Estimation: Predicting Problem-Solving Diversity in Interactive Environments.

Michael Eagle, Drew Hicks, and Tiffany Barnes
North Carolina State University, Department of Computer Science
890 Oval Drive, Campus Box 8206
Raleigh, NC 27695-8206
{mjeagle, aghicks3, tmbarnes}@ncsu.edu

ABSTRACT

Intelligent tutoring systems and computer aided learning environments aimed at developing problem solving produce large amounts of transactional data which make it a challenge for both researchers and educators to understand how students work within the environment. Researchers have modeled student-tutor interactions using complex networks in order to automatically derive next step hints. However, there are no clear thresholds for the amount of student data required before the hints can be produced. We introduce a novel method of estimating the size of the unobserved interaction network from a sample by leveraging Good-Turing frequency estimation. We use this estimation to predict size, growth, and overlap of interaction networks using a small sample of student data. Our estimate is accurate in as few as 10–30 students and is a good predictor for the growth of the observed state space for the full network, as well as the subset of the network which is usable for automatic hint generation. These methods provide researchers with metrics to evaluate different state representations, student populations, and general applicability of interaction networks on new datasets.

1. INTRODUCTION

Data-driven methods to provide automatic hints have the potential to substantially reduce the cost associated with developing tutors with personalized feedback. Modeling the student-tutor interactions as a complex network provides a platform for researchers to automatically generate next step hints. An *Interaction Network* is a complex network representation of all observed student and tutor interactions for a given problem in a game or tutoring system. In addition to their usefulness for automatically generating hints, interaction networks can provide an overview of student problem-solving approaches for a given problem.

Data-driven approaches cannot reliably produce feedback until sufficient data has been collected, a problem often referred to as the Cold Start problem. The precise amount of data needed varies by problem and environment. However, some properties of Interaction Networks allow us to estimate how much data is needed. Eagle et al. explored the structure of these student interaction networks and argued that networks could be interpreted as an empirical sample of student problem solving [5]. Students employing similar problem-solving approaches will explore overlapping areas of the Interaction Network. The more similar a group of students is, the smaller the overall explored area of the interaction network will ultimately be. Since we expect different populations of students to have different interaction networks, and different domains to require varying amounts of student data before feedback can be given, good metrics for the current and predicted quality of Interaction Networks are important.

In this work, we adapt Good-Turing frequency estimation to interaction level data to predict the size, growth, and "hintability" of interaction networks. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects [8]. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. In our context, network states (vertices) are the object types, and the student interactions (edges) leading to those states are observations.

We present several metrics, derived from Good-Turing frequency estimation. Our hypotheses are that these metrics: **H1:** Predict the probability that a student interaction will result in a state which was not previously observed **H2:** Describe the proportion of the network that has been observed for a population **H3:** Predict the expected size and growth of an interaction network when additional student data is added **H4:** Provide a quantitative comparison of different state representations for their ability to represent greater proportions of the network **H5:** Are useful for comparing different populations of users in how they explore the problem space

Additionally, we use the metrics to explore the subset of the interaction network that is useful for providing automatically generated hints. This provides us with estimates of the size, growth, and coverage of automatically generated hints. We find that our metrics quickly become accurate after collecting a sample of about 10 students. This has value as a metric to compare the quality of the interaction networks,

and will aid future researchers in determining an adequate state representation. We also show how two experimental groups, despite having the same amount of network coverage, have substantially different numbers of unique states. This supports previous work, suggesting that different populations of students produce different interaction networks [5], which has broad implications for generating hints as well as using the networks to evaluate student behavior.

1.1 Previous Work

Creation of adaptive educational programs is costly. This is, in part, because developing content for intelligent tutors requires multiple areas of expertise. Content experts and pedagogical experts must work with tutor developers to identify the skills students are applying and the associated feedback to deliver [13]. In order to address the difficulty in authoring intelligent tutoring content, Barnes and Stamper built an approach called the Hint Factory to use student data to build a Markov Decision Process (MDP) of student problem-solving approaches to serve as a domain model for automatic hint generation [18]. Hint Factory has been applied in tutoring systems and educational games across several domains [7, 14, 6], and been shown to increase student retention in tutors [19].

Early work with the Hint Factory method used a Markov Decision Process constructed from students' problem-solving attempts. Eagle and Barnes further developed this structure into a complex network representation of student interactions with the system, called an *Interaction Network* [5]. Complex networks are graphs or networks which contain non-trivial topological features unlikely to appear in simple or random networks. The Interaction Network representation can be used as a visualization of student work within tutors. The effectiveness of Interaction Networks as visualizations was shown by Johnson et al. who created a visualization tool *InVis* to aid instructors in analyzing student-tutor data [11].

Other approaches to automated generation of feedback have attempted to condense similar solutions in order to address sparse data sets. One such approach converts solutions into a canonical form by strictly ordering the dependencies of statements in a program [15]. Another approach compares *linkage graphs* modelling how a program creates and modifies variables, with nested states created when a loop or branch appears in the code [10]. In the Andes physics tutor, students may ask for hints about how to proceed. Similarly to Hint Factory-based approaches, a solution graph representing possible correct solutions to the problem was used. However their solution space was explored procedurally rather than being derived from student data, and they used plan recognition to decide which of the problem derivations the student is working towards [20].

Interaction networks are scale-free networks. This is a property of complex networks whose degree distribution is heavy-tailed, often a power law distribution. In practice, this means that a few vertices have degree that is much larger than the average, while many vertices have degree somewhat lower than average [5]. Eagle et al. argued that students with similar problem solving ability and preferences would travel into similar parts of the network, resulting in some states being more important to the problem than others [5]. Using these "hub" states, sub-regions of the network corresponding to high-level approaches to the problem were derived. These sub-regions captured problem-solving differences between two experimental groups [4].

2. METHODS AND MATERIALS

For the purposes of this work, we are using datasets from three different environments to build our interaction networks. Summaries of these datasets are found in Table 1. The first dataset is from the Deep Thought tutor, used in previous work by Stamper et al. [19]. This dataset was collected for a between groups experiment investigating the use of data-driven hints, so we split the dataset into two groups, DT1-C, the control group from that experiment, and DT1-H, the group that received hints. We selected this dataset to explore and evaluate H5.

The second dataset comes from the game BOTS. Here, we have the same students and interactions represented in two different ways: First. using *codestates* (the programs users wrote) and second using *worldstates* (the output of those programs). The advantages and disadvantages of these state representations were explored in previous work by Peddycord and Hicks [14]. We split this dataset into two groups as well (BOTS-C and BOTS-W) one for each state representation used. We selected this dataset for evaluation of H4.

Our third and largest dataset comes from an updated version of the Deep Thought tutor, called Deep Thought 3. Unlike with the other datasets, Deep Thought 3 features an AI problem selection component [12]. This means that not all students will have had access to all problems. In addition, there is a larger number of problems in this dataset. We selected this dataset, as the larger number of problems effectively splits student data across multiple networks. H1–H3 are relevant towards measuring the quality of networks produced for new problems.

Table 1: Dataset summary: the total number of students in the dataset, the number of distinct problems, and the average number of students represented in each network.

Dataset	Total N	Num Problems	Mean Net N
DT1-H	203	11	83.73
DT1-C	203	11	63.82
DT3	341	59	78.41
BOTS-C	125	12	99.75
BOTS-W	125	12	99.75

2.1 Constructing an Interaction Network

An *Interaction Network* is a complex network representation of all observed student and tutor interactions for a given problem in a game or tutoring system. To construct an Interaction Network for a problem, we collect the set of all solution attempts for that problem. Each solution attempt is defined by a unique user identifier, as well as an ordered sequence of interactions, where an interaction is defined as {initial state, action, resulting state}, from the start of the

problem until the user solves the problem or exits the system. The information contained in a *state* is sufficient to precisely recreate the tutor's interface at each step. Similarly, an *action* is any user interaction which changes the state, and is defined as {action name, pre-conditions, post-conditions}. In Deep Thought, for example, an action would be the logical axiom applied, the statements it was applied to, and the resulting derived statement. Figure 1 displays two Deep Thought interactions. The first interaction works forward from STEP0 to STEP1 with action $SIMP$ (simplification) applied to $(Z \wedge \neg W)$ to derive $\neg W$. The second interaction works backward from STEP1 to STEP2 with action $B - ADD$ (backwards addition) applied to $(X \vee S)$ to derive the new, unjustified statement S.

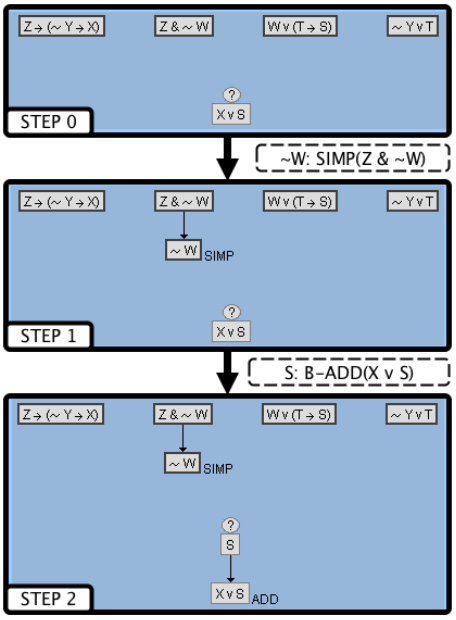


Figure 1: Example of state to state transitions within the Deep Thought (DT1) propositional logic tutoring system.

Once the data is collected. we use a *state matching function* to combine similar states. In Deep Thought, we combine states that consist of all the same logic statements, regardless of the order in which those statements were derived. This way, the resulting state for a step STEP0, STEP1, or STEP2 in Figure 1 is the set of justified and unjustified statements in each screenshot, regardless of the order that each statement was derived. In BOTS, two state matching functions were used: one which combined states based on the code in students' programs, and another which instead used the output of those programs. Similarly, we use an action matching function to combine actions which result in similar states, while preserving the frequency of each observed interaction.

2.2 Providing Hints

Stamper and Barnes' Hint Factory approach generates a next step Hint Policy by modeling student-tutor interactions as a Markov Decision Process [18]. This has been adapted to work with interaction networks by using a Value Iteration algorithm on the states [5]. We generate a graph of all student interactions, combining identical states using a state matching function. Then, we calculate a fitness value for each state. We assign a positive value (100) to each goal state, that is a state configuration representing a solution to the problem. We assign an error cost (-5) for error states. We also assign a small cost to performing any action, which biases hint-selection towards shorter solutions. We then calculate fitness values $V(s)$ for each state s, where $R(s)$ is the initial fitness value for the state, γ is a discount factor, and $P(s, s')$ is the observed frequency with which users in state s take an action resulting in state s'. After this, we use value iteration [2] to repeatedly assign each state a value based on its neighbors and action costs, weighted by frequency.

After applying this algorithm, we can provide a hint to guide the user toward the goal by selecting the child state with the best value. We can do this for any observed state, provided that a previous user has successfully solved the problem after visiting that state. In the original work with Hint Factory on the Deep Thought tutor, the algorithm was permitted to backtrack to an earlier state if it failed to find a hint from the current state. However, not all environments allow the user to backtrack and there are risks of the backtracking hints to provide irrelevant information. Because of this inconsistency across domains, we did not permit backtracking for the purposes of the comparisons in this paper.

We define a state, S to be *Hintable* if S lies on a path which ends at a goal state. We define the *Hintable* network to be the subset of the interaction network containing only *Hintable* states and edges between hintable states; That is, the induced subgraph on the set of *Hintable* states.

2.3 Cold Start Problem

Barnes and Stamper [1] approached the question of how much data is needed to get a certain amount of overlap in student solution attempts by incrementally adding student attempts and measuring the step overlap over a large series of trials. This was done with the goal of producing automatically generated hints, and solution attempts that did not reach the goal were excluded. Peddycord et al. [14] used a similar technique to evaluate differences in overlap between two different interaction network state representations.

The "Cold Start problem" is an issue that arises in all data-driven systems. For early users of the system, predictions made are inaccurate or incomplete [17, 16]. If there are insufficient data to compare to (not enough user ratings, or not enough student attempts) then the quality of the recommendations suffers and in some cases no recommendation can be provided. The term is commonly used in the field of collaborative filtering and recommender systems, but it can be used to describe three related issues, the "new user," the "new item," and the "new community" [3] Cold Start problems. The "new user" problem refers to the difficulty of making recommendations to a user who has performed no actions. The "new item" problem refers to the difficulty of suggesting users visit a newly added, unobserved state. The new community Cold Start problem refers to situations where not enough observations exist to make recommendations for new users. The "new community" definition corresponds most closely to the difficulty of generating hints for

an entirely new problem in an intelligent tutoring system or educational game.

To measure our ability to address this problem, we add all interactions from a single student, one at a time, to the interaction network. This is in order to simulate the growth of the network. We repeat this process for each student, measuring the performance of our model each time. We measured the proportion of currently observed states to total observed states for the entire data set, as well as for the subset of states from which a goal is reachable. To control for ordering effects, we repeated this trial 1000 times using a different random ordering of students each time, and aggregated the results.

2.4 Good-Turing Network Estimation

We present a new method for estimating the size of the unobserved portion of a partially constructed Interaction Network. Our estimator makes use of Good-Turing frequency estimation [8]. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. Gale and Sampson revisited and simplified the implementation [8]. In its original context, given a sample text from a vocabulary, the Good-Turing Estimator will predict the probability that a new word selected from that vocabulary will be one not previously observed.

The Good-Turing method of estimation uses the frequency distribution, the "frequency of frequencies," from the sample text in order to estimate the probability that a new word will be of a given frequency. Based on this distribution, the probability of observing a new word in an additional sample is estimated with the observed proportion of words with frequency one. This estimate of unobserved words is used to adjust the probabilities of encountering words of frequencies greater than one.

We adapt the Good-Turing Estimator to interaction networks by using the states with an observed frequency of one to estimate the proportion of "frequency zero" states. Interaction networks represent the observed interactions and therefore we also use this value to estimate the probability that a new interaction will transition into a new state. We use P_0 as the expected probability of the next observation being an unseen state. P_0 is estimated by:

$$P_0 = \frac{N_1}{N} \tag{1}$$

Where N_1 is the total number of frequency 1 states, and N is the total number of interaction observations. Since N_1 is the largest group of states, the observed value of N_1 is a reasonable estimate of P_1. P_0 can then be used to smooth the estimation proportions of the other states. The proportion of states with observed frequency r is found by:

$$P_r = \frac{(r+1)S(N_{r+1})}{N} \tag{2}$$

where $S()$ is a smoothing function that adjusts the value for large values of r [8].

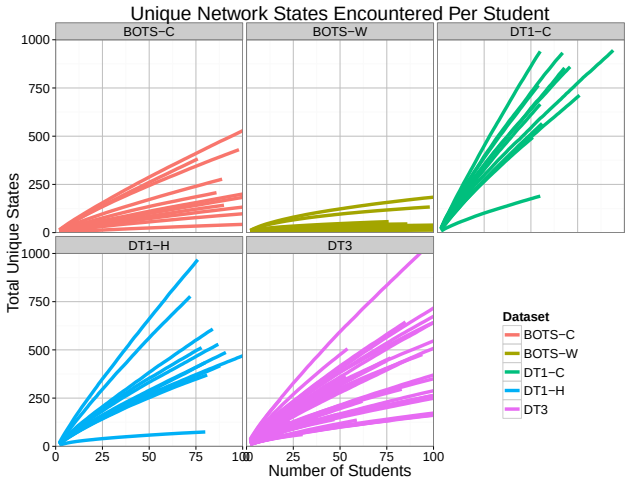


Figure 2: The growth of new states as new students are added for each problem, for each dataset.

Our version of P_0 is the probability of encountering a new state (a state that currently has a frequency of zero,) on a new interaction. We also interpret this as the proportion of the network missing from the sample. We will refer to an interaction with a unobserved state as having *fallen off* of the interaction network. We will use the complement of P_0 as the estimate of *network coverage*, I_C, the probability that a new interaction will remain on the network: $I_C = 1 - P_0$.

The *state space* of the environment is the set of all possible state configurations. For both the BOTS game and the Deep Thought tutor the potential state space is infinite. For example, in the Deep Thought tutor a student can always use the addition rule to add new propositions to the state. However, as argued in Eagle et. al. [5], the actions that reasonable humans perform is only a small subset of the theoretical state space; the actions can also be different for different populations of humans. We will refer to this subset as the *Reasonable State Space*, with *unreasonable* being loosely defined as actions that we would not expect a human to take. An interaction network is an empirical sample of the problem solving behavior from a particular population, and is a subset of the state space of all possible *reasonable* behaviors. Therefore, our metrics P_0 and I_C are estimates of how well the observed interaction network represents the reasonable state space.

3. RESULTS

In order to evaluate the performance of the unobserved network estimator, P_0, and the network coverage estimator, I_C, for each problem in each of our 5 datasets we randomly added students from the sample, one at a time until all student data had been included. At each step, T, we recorded the values of our estimators using only the data that had been encountered up until then. This simulates a real world use-case, where additional students are added over time. We repeated this process 1000 times and averaged the results. Figure 2 shows the growth of unique states as students are added for the interaction networks generated by each problem (line) in each of the five datasets.

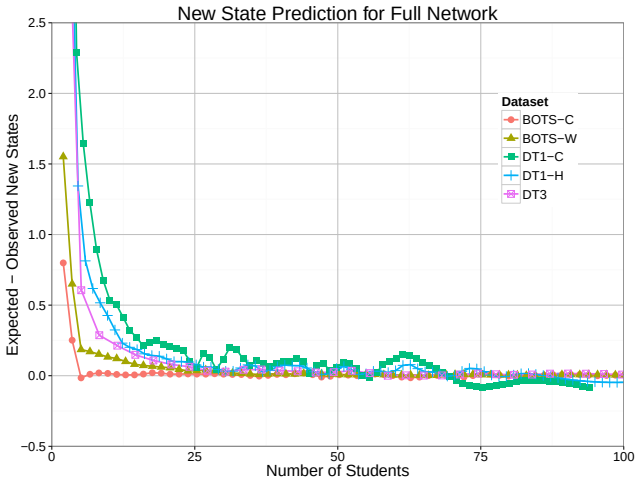


Figure 3: The average absolute error between the estimated number of new states and the observed new states over the number of students for all problems in each of the four datasets. P_0 accurately predicts the observed values after roughly 10 students, rarely being off by more than one after that.

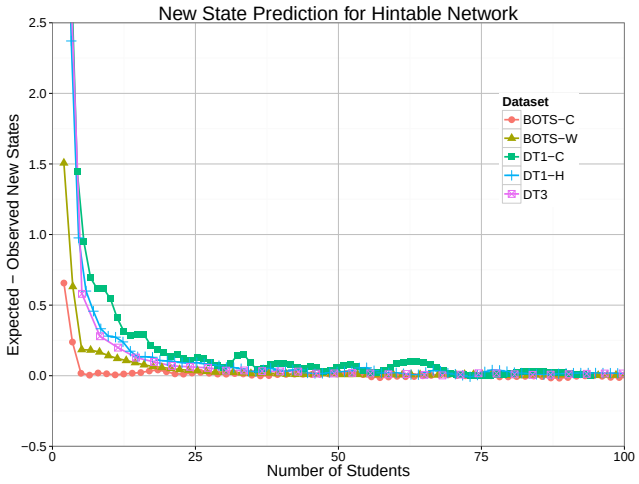


Figure 4: For the hintable states, the average difference between the estimated number of new states and the observed new states over the number of students for all problems in each of the four datasets. P_0 accurately predicts the observed values after roughly 10 students, rarely being off by more than one after that.

3.1 H1: Prediction of New States

In order to evaluate P_0 for the prediction of new states (states that are frequency $= 0$ on time T_i, but will be frequency $= 1$ on T_{i+1}. At each T we add an additional student and compare the expected number of frequency 1 states, E_{S1}, vs. the observed number, O_{S1}. Across all five datasets, Figure 3 shows the differences between the expected and observed number of new states. The $P_O \times Interactions$ prediction for new states follows closely with the observed number, the estimates increase in accuracy rapidly over the first ten students and are rarely off by more than a fraction of a state afterwards. Figure 4 shows the results of running this process on only the hintable portion of the interaction network for each data set.

3.2 H2: Network Coverage

We have defined network coverage I_C as the proportion of interactions which lie within the previously observed network. Another interpretation is that I_C is the probability of an interaction resulting in a state that has been previously observed. This value is the complement of P_0. Figure 5 and 7 display the results of network coverage and its growth as additional students are added.

3.3 H3: Predicting Future Network Size

In order to further evaluate the use of P_0 and I_C we calculated a prediction for the final size of the network, given the number of students in each dataset, at each time stamp. The equation for this prediction is:

$$|V(IN)| = (NewSample * P_0) + U_T. \quad (3)$$

Where $|V(IN)|$ is the number of unique vertices (states) in the final network, *NewSample* is the number of new interactions added, P_0 is the estimation of new states added, and U_T is the number of unique states observed at time T. The results are averaged across all problems for each dataset and are presented in figures 8 and 9. This prediction rapidly improves and after roughly 20% of the sample is added, can accurately predict the final number of unique states for the network. This combined with the accuracy of P_0 reveals the short term and long term accuracy for the estimator.

3.4 H4: Comparing State Matching Functions

The network coverage metric, I_C, allows an easy method of estimating the differences in state matching functions and student network overlap. We can use I_C with two potential matching functions, and get an estimate of the remaining network, to quickly compare different potential state representations as well as to find a state generalization that will allow for a desired amount of network coverage.

The estimate based on the above methods has proven useful for comparing State Matching functions to help determine which produces more relevant hints. Figure 6 shows the BOTS interface, with the user's program (codestate) and the game world (worldstate) both illustrated. In previous work investigating the Cold Start problem on the BOTS data set, we measured "coverage" in terms of how much of the newly added test data was already present in the training set [9, 14]. Compare this analysis to Figure 5 which shows the estimated probability that a student's next action will result in an observed state, I_C. After 100 students, the probability that a student will generate a new *codestate* is still quite high, $P_0 > .25$. In comparison, after the same number of students, the probability of generating a new *worldstate* is extremely low, $P_0 < .02$. This result supports both our intuition and our results from the previous work, that students will continue to generate new *codestates*, but that these different *codestates* will collapse to previously observed *worldstates*.

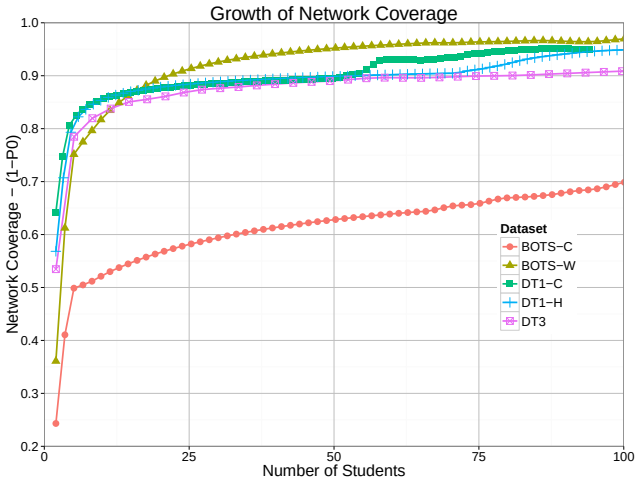


Figure 5: The estimated network coverage I_C for each of the 5 datasets, note the poor coverage for the BOTS-C dataset. The BOTS-W state is more general and has the much higher coverage.

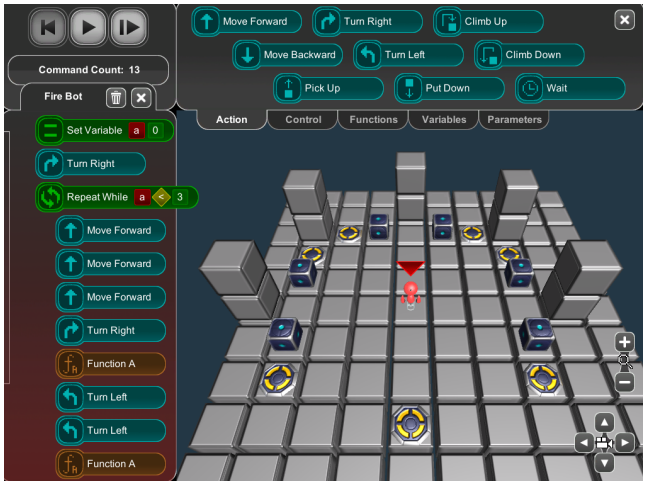


Figure 6: An image of the main gameplay interface for BOTS. The left hand side of the screen shows the user's program, used to derive code states. The right-hand side shows the game world, where the program output determines the world states.

3.5 H5: Comparing Populations

Samples from different populations have different resulting interaction networks. The size of the represented network can tell us about the similarity of student approaches in the sample. If students are more alike in the types of actions they perform, fewer students will be needed to achieve a similar amount of overlap. We can also see that adding students from a dissimilar population will not always increase estimated network coverage (I_C), and can potentially decrease it. This has implications about the importance of building hints for one population and applying it for another. In other work we have already shown that different groups are likely to visit different parts of the networks [4]. Here we expand on that analysis by showing that the two

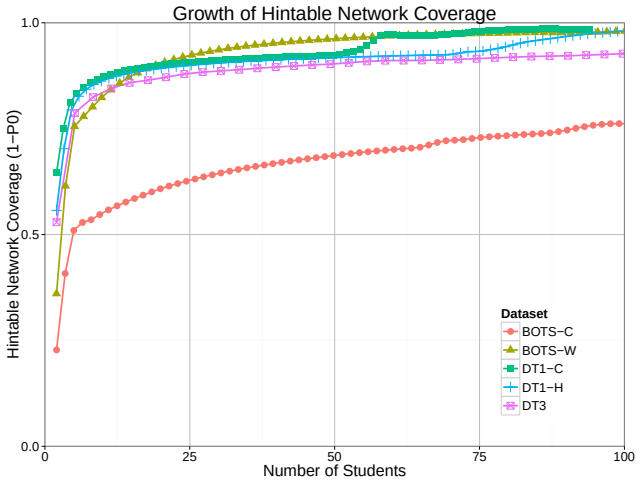


Figure 7: For the hintable network: the estimated network coverage I_C for each of the 5 datasets. Even the lowest performing hint network BOTS-C reaches roughly 70% coverage by 100 students.

Table 2: Different populations have different spread in problem exploration.

Group	P_0	States	Interactions	F_1
Hint	0.09	514.61	2709.84	250.09
Control	0.10	720.12	3904.92	340.00

groups, while having the same amount of network coverage, have a different number of unique states. Table 2 shows the results between the Hint group, which received hints on a subset of the problems, and the Control group which never received hints. This corresponds with results from Eagle et al. [4] in which they uncovered significant differences in the student overall approaches. This result adds to that an estimation of how complete each network was, revealing that additional data was not likely to change the result. It also shows some evidence for a *trail blazing effect*. When provided hints, students collectively explore a smaller area of the state space.

3.6 Estimating the effect of filtering

Visualizations must struggle with an "information to ink" ratio. There is a trade-off between displaying full information and overwhelming the viewer, and displaying only the most frequent states and potentially misleading the viewer by eliminating information. *InVis*, a visualization tool for exploring Interaction Networks allowed users to filter by frequency[11]. We can use the Good-Turing Estimation to calculate the amount of information removed by filtering frequency of a certain degree. P_0 is the proportion of the network missing, $I_{C_{>r}} = I_C - P_{1--r} + P_0$, where r is a threshold value for removing low frequency states, and $P_1 - -r$ is the sum of P_1 through P_r. This should be a useful metric for visualizations for measuring the amount of network that is hidden by filtering. It is also useful to show that sometimes a large number of graphical elements can be removed, with only a small amount of interaction information lost.

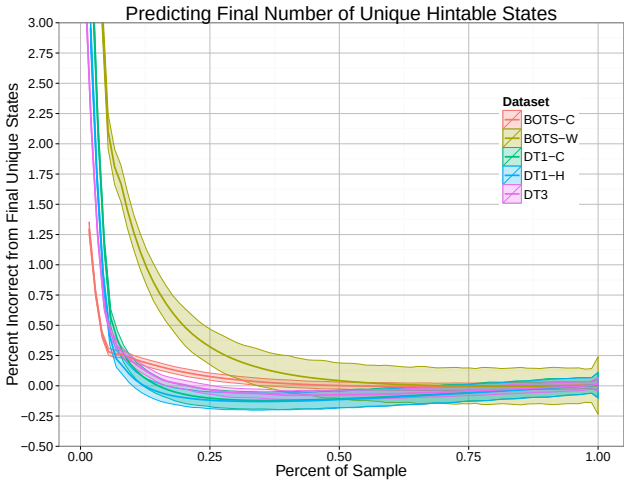


Figure 8: Prediction of total final number of states, as observed number of states increases. Note that for small t, the estimate is very high (up to 300% over prediction), but becomes fairly accurate after roughly 20% of the sample is measured.

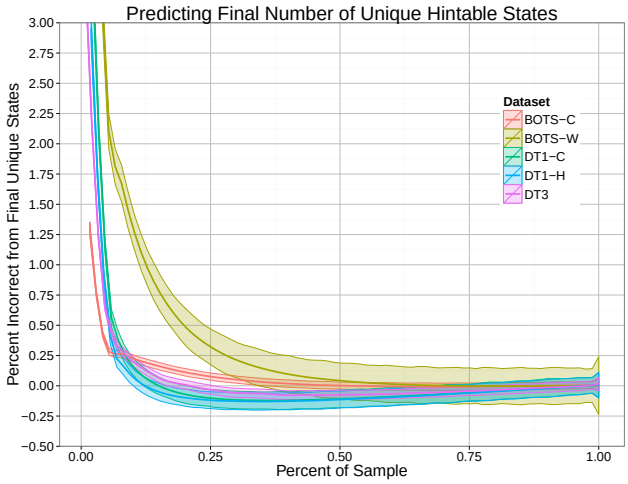


Figure 9: Prediction of total final number of goal states, as observed number of states increases. Note that for small t, the estimate is very high, but becomes an underestimate as t increases. P_0 can predict the number of additional hintable states that can be added for a additional sample of data.

4. DISCUSSION

Good-Turing Estimation works well in the contexts of interaction networks. We were able to provide an easily calculable estimate of the proportion of the network not yet observed P_0. This value alone is a useful high level metric for the percentage of times a student interaction results in a previously unobserved state. The P_0 score for the hintable network is likewise an estimate of the probability that a student will "fall off" of the network from which we can provide feedback. Our network coverage metric I_C allows a quick and easy to calculate method of comparing different state representations, as well as quantifying the difference. We believe that this metric can replace the commonly used cold start method of evaluating the "hintability" of a network. I_C is also valuable to quickly gauge the applicability of a new domain to interaction networks. The majority of the calculations can be performed on the transactional data. The growth trends for our five datasets were often clear after only ten students.

Our network estimators also have implications given our previous theories on the network being a sample created from biased (non-random) walks on the problem-space, as the more homogeneous the biased walkers are, the faster the network will represent the population and the fewer additional states will be explored. We revisited our previous results [4], and found that students with access to hints explored less overall unique states. This implies that the students were more similar to each other in terms of the types of actions and states they visited within the problem. Overall, this result supports the idea that different populations of students will have different interaction networks. The implications of this for generating hints are great. Building hints on one population might not work as well in another, and adding interventions or hints can dramatically reduce the number of states visited by the students. Future work should explore the possibility of having multiple network representations and choosing to match the student with the one closely resembling them.

As you can see in figure 8, our estimator starts out drastically overestimating the number of unobserved states in the network. As we collect data, this eventually becomes a slight underestimate, eventually converging on the correct number of states. One explanation for why this might be the case is the method by which undiscovered states are added to the network. By using this model for our estimator, we are making an assumption that states are selected independently of one another. At the beginning, when data is sparse, this assumption is not particularly harmful, since undiscovered states are relatively common. However, as our dataset becomes richer, we underestimate the probability of adding an unobserved state because we do not take into account the effect of "trail-blazing" which increases the probability of adding additional unobserved states after the first. Eagle and Barnes found that interaction networks had properties of scale-free networks. [5]. In particular, their degree distributions follow a power law, with a few vertices having much higher degree than the average for the network. It is likely that taking into account the scale-free and hierarchical nature of the networks will provide methods to improve on our estimators.

5. CONCLUSIONS AND FUTURE WORK

We have adapted Good-Turing frequency estimation for use with networks built from student-tutor interactions. We found that the estimator for the missing proportion of the network P_0 was accurate in predicting the number of new states discovered with new data. We also found that we could accurately measure network coverage with I_C for both the regular network, as well as the network of hintable states. This provides us with a metric to compare different state representations as well as determine the suitability of inter-

action network methods to different tutoring environments. We were also able to use these metrics to provide accurate predictions for the size of networks expected given more data samples, which will be useful for predicting the amount of additional data needed to provide a desired amount of hintable network coverage. Finally, we used the estimate of network coverage to compare different student populations to show that the addition of hints in one environment had an effect on the number of states explored by students.

Future work will include expanding on these *global* measures of the network and exploring *local* measures of coverage. Rather than compute coverage for the entire network we can use methods such as approach map regioning [4] to find meaningful sub-networks and calculate the metrics for those. The region level values of P_0 can estimate the "riskyness" of certain approaches to the problem. The I_C metric can direct attention to parts of the network that are not well explored, perhaps allowing additional hints to be obtained by starting advanced users in those areas.

6. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. #0845997, #1432156, #1015456, #0900860 and #1252376.

7. REFERENCES

[1] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)*, pages 373–382, 2008.

[2] R. Bellman. A markovian decision process. Technical report, DTIC Document, 1957.

[3] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26(0):225 – 238, 2012.

[4] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. *Proceedings of the Seventh International Conference on Educational Data Mining*, 2014.

[5] M. Eagle, D. Hicks, P. III, and T. Barnes. Exploring networks of problem-solving interactions. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.

[6] M. Eagle, M. Johnson, T. Barnes, and A. K. Boyce. Exploring player behavior with visual analytics. In *FDG*, pages 380–383, 2013.

[7] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 491–498, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[8] W. A. Gale and G. Sampson. Good-turing frequency estimation without tears*. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

[9] A. Hicks, B. Peddycord III, and T. Barnes. Building games to learn from their players: Generating hints in a serious game. In *Intelligent Tutoring Systems*, pages 312–317. Springer, 2014.

[10] W. Jin, T. Barnes, J. Stamper, M. J. Eagle, M. W. Johnson, and L. Lehmann. Program representation for automatic hint generation for a data-driven novice programming tutor. In *Intelligent Tutoring Systems*, pages 304–309. Springer, 2012.

[11] M. W. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks.

[12] B. Mostafavi, M. Eagle, and T. Barnes. Towards data-driven mastery learning. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.

[13] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[14] B. Peddycord III, A. Hicks, and T. Barnes. Generating hints for programming problems using intermediate output.

[15] K. Rivers and K. R. Koedinger. Automating hint generation with solution space path construction. In *Intelligent Tutoring Systems*, pages 329–339. Springer, 2014.

[16] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[17] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[18] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197–201, 2008.

[19] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2013.

[20] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.

Why do the rich get richer? A structural equation model to test how spatial skills affect learning with representations

Martina A. Rau
Department of Educational Psychology
University of Wisconsin—Madison
1025 W. Johnson St
Madison, WI 53706
+1-608-262-0833
marau@wisc.edu

ABSTRACT
Spatial skills predict students' success in STEM domains. This paper aims to better understand the difficulties of students with low spatial skills in using interactive graphical representations. I present a mediation analysis with test and log data from 117 students who worked with an intelligent tutoring system for chemistry. The analysis is based on (1) a knowledge component model that describes knowledge students acquire as they solve problems with graphical representations, (2) a search for features that describe students' interactions with the representations and that are predictive of students' learning gains, and (3) a structural equation model that tests whether these features statistically mediate the effect of spatial skills on students' learning gains. Results show that only students' ability to plan representations before they construct them mediates the effect of spatial skills on learning gains. This finding suggests that these students may need more support *before* they construct representations.

Keywords
Spatial skills, intelligent tutoring systems, interactive representations, STEM learning.

1. INTRODUCTION
Students' spatial skills predict learning success in STEM domains [1, 2]: students with low spatial skills tend to show lower achievements in STEM domains and they are less likely to pursue careers in these domains. Spatial skills are important for STEM learning because many concepts in STEM domains are inherently visuo-spatial. For example, astronomers have to visualize the solar system, engineers have to visualize interactions among components of a machine, and chemists have to visualize movements of atoms and electrons. To make these concepts accessible to students, instructional materials in STEM domains tend to heavily rely on the use of graphical representations [5, 6]. Graphical representations are external representations that use visuo-spatial features to depict domain-relevant concepts (as opposed to text or symbols). As a consequence, students have to make sense of visuo-spatial relationships depicted by graphical representations to understand abstract concepts in STEM domains [7].

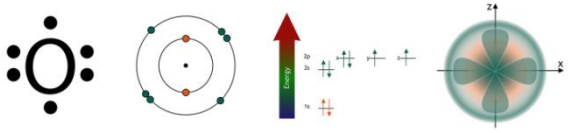


Figure 1. Graphical representations of an oxygen atom: Lewis structure, Bohr model, energy diagram, orbital diagram.

Consider, for example, a student who is learning about atomic structure. Figure 1 shows the graphical representations that instructional materials typically use to illustrate atomic structure [8]. Lewis structures (left) show paired an unpaired valence electrons, Bohr models (center-left) show all electrons in atomic shells, energy diagrams (center-right) depict electrons in orbitals with their energy level, and orbital diagrams (right) show the spatial arrangement of non-empty orbitals. To understand atomic structure, students have to integrate the information depicted in these graphical representations into a visuo-spatial mental model of how electrons are arranged relative to the atom's nucleus, and how they move according to probabilistic laws.

Integrating such information into a mental model of the domain-relevant concepts requires students to hold the relative location of the depicted objects in working memory and to mentally rotate these objects [9]. The cognitive load imposed by this task is arguably higher for students with low spatial skills than for students with high spatial skills [1]. As a consequence, students with low spatial skills may fail at this task, which might jeopardize their learning success [1, 5, 9]. On the flip side, students with high spatial skills are more successful at integrating visuo-spatial information into mental models, and—consequently—are likely to show higher learning gains. Thus, the rich (in spatial skills) get richer (in content knowledge).

Educational technologies such as intelligent tutoring systems (ITSs) hold particular promise for breaking the "the-rich-get-richer" rule and for creating an "everyone-gets-richer" rule, because they can address the needs of students with low spatial skills in several ways. First, ITSs can provide interactive tools that students can use to construct representations while receiving assistance and feedback. Such support for learning with interactive graphical representations can enhance learning outcomes [10], in particular for students with low spatial skills [11]. Second, ITSs have the capability to provide individualized support that adapts to student characteristics [12]. Adapting instructional support to the individual student's spatial skills has been shown to improve their spatial skills [13] as well as their learning of content knowledge [14].

However, before we can design ITSs that tailor support for using interactive representations to the needs of students with low spatial skills, we first have to understand what makes this learning task difficult for these students. This paper presents a first step towards this goal. Specifically, this paper investigates the following two questions: (1) Which aspects of problem solving with interactive graphical representations are more difficult for students with low spatial skills than for students with high spatial skills? (2) Which of these difficulties explain why students with

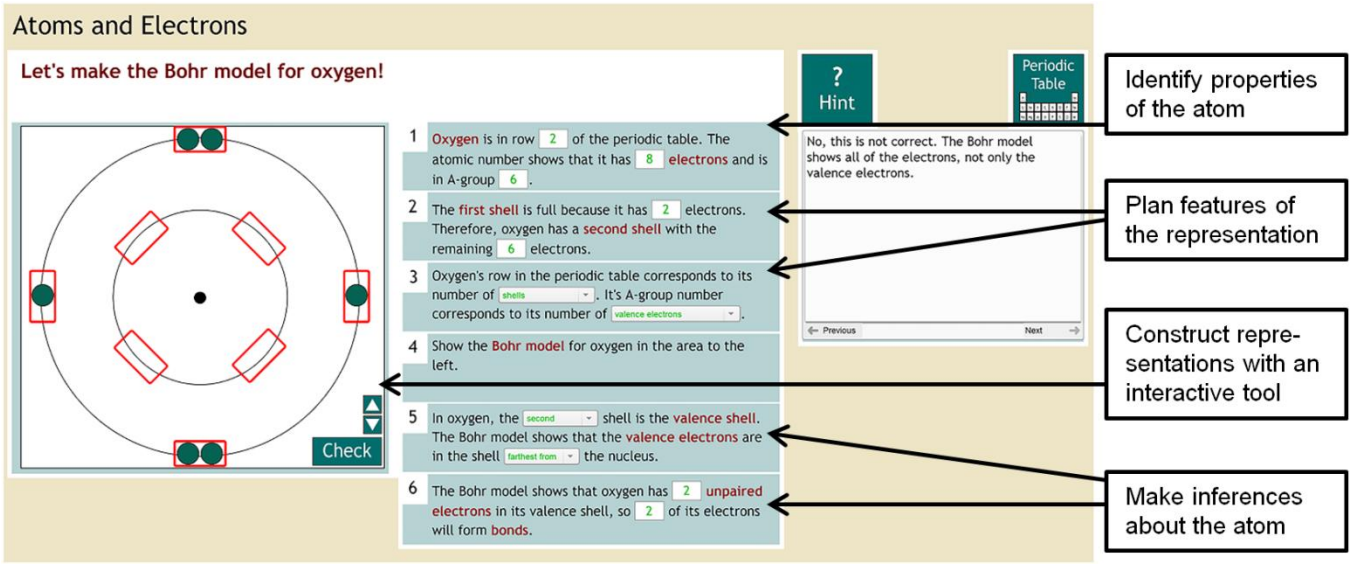


Figure 2. Example screen shot of a tutor problem: students construct a Bohr model of oxygen.

low spatial skills have lower learning outcomes in chemistry? To address these questions, I conducted a mediation analysis that tested which aspects of students' problem-solving performance account for the effect of spatial skills on learning outcomes. The mediation analysis was carried out with a data set obtained from an experiment with an ITS for chemistry learning in which students had to use interactive tools to construct graphical representations of atoms.

2. CHEM TUTOR

The data set used in this paper was obtained from an experiment with Chem Tutor: an ITS for undergraduate chemistry [15]. The goal of Chem Tutor is to enhance learning by helping students understand graphical representations of abstract concepts [16]. Chem Tutor targets foundational concepts of introductory undergraduate courses, such as atomic structure and bonding. The design of Chem Tutor is based on surveys with undergraduate chemistry students and instructors, interviews and eye-tracking studies with undergraduate and graduate students, and extensive pilot testing in the lab and the field [15]. Chem Tutor was built with Cognitive Tutor Authoring Tools [17], which facilitates rapid iterations of prototyping and pilot-testing involved in such user-centered design approaches.

In the present experiment, students worked with the atoms and electrons unit of Chem Tutor. This unit features interactive tools that students use to construct a variety of graphical representations of atoms: Lewis structures, Bohr models, energy diagrams, and orbital diagrams (see Figure 1). The tutor problems are structured as follows. First, students are prompted to think about the properties of the atom. They can use the periodic table to look up information about the atom (e.g., oxygen has eight electrons). Second, students are prompted to plan what the given representation will look like (e.g., the Bohr model of oxygen should show two shells). Third, students use an interactive tool to construct the representation of the given atom. Students receive error-specific feedback on their interactions (e.g., "The Bohr model shows all of the electrons, not only the valence electrons"). Students have to construct a correct graphical representation before they can continue. Fourth, students are prompted to make inferences from the given graphical representation about the atom (e.g., the number of valence electrons allow to approximate the number of bonds the atom forms). Figure 2 shows an example tutor problem in which students construct the Bohr model of an oxygen atom. The interface of the problems builds up step-by-step, as shown in Figure 3.

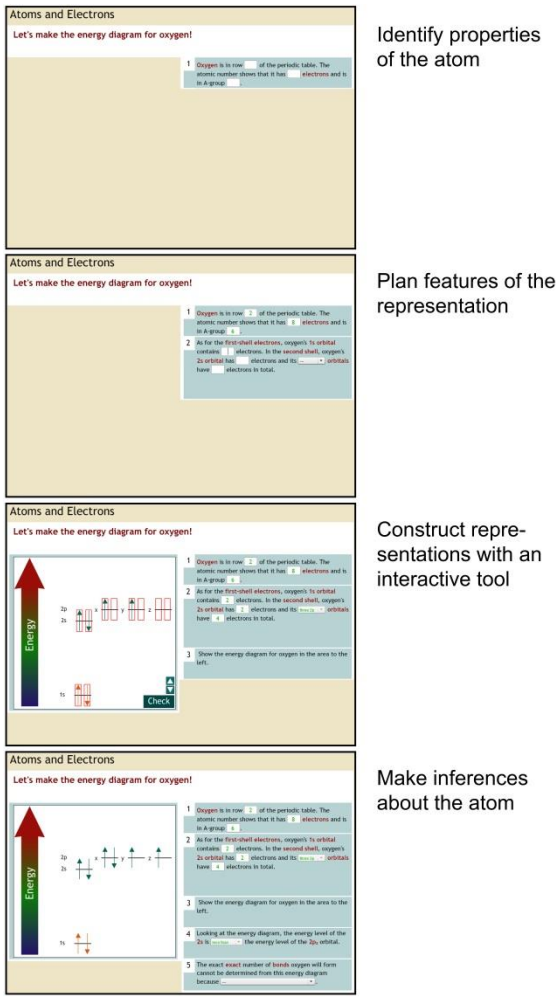


Figure 3. Sequence of screen shots showing how the interface updates step by step as students construct an Energy diagram.

3. EXPERIMENT

The experiment investigated whether Chem Tutor helps undergraduate students learn chemistry. For a detailed description of the experiment, refer to [18].

3.1 Participants

117 undergraduate students from a university in the mid-western United States participated in the experiment. 79% of the students were enrolled in general chemistry for non-science majors. According to the instructor of this course, these students had no experience with the graphical representations used in the Chem Tutor unit, with the exception of the common Lewis structure. 13.4% of the students were enrolled in general chemistry for science majors, 2.5% were enrolled in advanced general chemistry. According to the instructors of these courses, these students had experience with all graphical representations used in the Chem Tutor unit. The remaining 5% of the students were not currently enrolled in a chemistry course.

3.2 Assessments

Students' chemistry knowledge was assessed three times: before they started working with Chem Tutor (pretest), after they completed half of the tutor problems (intermediate posttest), and after they completed all tutor problems (final posttest). Three isomorphic test forms were used: they asked structurally identical questions but used different problems (e.g., with different atoms). The order in which students received the test forms was counterbalanced. The tests assessed reproduction and transfer of the chemistry content covered in Chem Tutor. Reproduction items used a format similar to the Chem Tutor problems. Transfer items asked students to apply the knowledge Chem Tutor covered in ways they had not been asked to do in the Chem Tutor problems. The tests included items with and without representations. In addition, spatial skills were assessed with the Vandenberg & Kuse mental rotation ability test [19]. This test presents students with a drawing of an object and asks them to identify which of four other drawings show the same object. This task requires spatial skills because students have to mentally rotate the given object to align it with the comparison objects. This test was chosen because it has been used in prior research on the impact of students' spatial skills on STEM learning [1, 2, 4, 5, 7].

3.3 Procedure

The experiment took place in the laboratory and involved two sessions of about 90 minutes each. Sessions were scheduled no more than three days apart. In session 1, students first completed the mental rotation test and the chemistry pretest. They then received an introduction into using Chem Tutor. Next, they worked through half of the problems in Chem Tutor's atoms and electrons unit. At the end of session 1, students took the intermediate chemistry posttest. In session 2, students worked through the remainder of the tutor problems. At the end of session 2, they took the final chemistry posttest. All students worked on the tutor problems at their own pace and were able to finish the assigned tutor problems in the available time.

3.4 Results

Results from the analysis of the test data show that there were significant learning gains on the chemistry knowledge test, $F(2,230) = 6.18$, $p < .01$. A regression of students' spatial skills on learning gains (i.e., performance on the posttest, controlling for pretest performance) showed that spatial skills were a significant predictor of learning gains ($\beta = .34$, $p < .01$), such that students with high spatial skills showed higher learning gains than students with low spatial skills.

4. OPEN QUESTIONS

The finding that students with lower spatial skills had lower learning gains as the result of an intervention that relies on graphical representations is not surprising: it aligns with prior research on the role of spatial skills in STEM learning [1, 4, 5, 9]. It is conceivable that working with interactive graphical representations requires students to make sense of how abstract properties of atoms can be translated into visuo-spatial elements of graphical representations. It is well documented that this is more difficult for students with lower spatial skills [1, 4, 5, 9].

A first question that remains thus far unanswered, however, is how these difficulties affect how students interact with tutor problems. There are several aspects of the problems in Chem Tutor that may be more difficult for students with low spatial skills. First, these students may struggle with the first part of the tutor problems: identifying properties of atoms. Students with low spatial skills may have trouble retrieving facts that describe properties of atoms because they cannot imagine what an atom looks like. They might also struggle in using resources such as the periodic table to retrieve this information. Second, students with low spatial skills may struggle with the planning part of the tutor problems, because this step requires them to think about how properties of an atom can be visualized. Third, it is possible that these students struggle more when constructing graphical representations because they have to translate text-based information into visuo-spatial elements of the graphical representations. Finally, it is possible that these students struggle more in using representations to make inferences about the atom because this requires them to imagine how the visualized properties determine dynamic behavior of electrons (e.g., electron movement) and of atoms (e.g., tendency to form bonds).

A second question that remains open is how these difficulties relate to learning gains. While it is possible that all of the aspects just described are more difficult for students with low spatial skills, some difficulties may play a larger role than others in explaining why these students show lower learning gains. Understanding which difficulties account for the fact that students with lower spatial skills show lower learning gains will enable us to provide more appropriate support for these students.

5. FEATURE SELECTION

To investigate why spatial skills predict students' learning gains as they work with interactive graphical representations, I used a structural equation model to conduct a mediation analysis. Structural equation models provide a unified framework to test mediation hypotheses, estimate total effects, and separate direct from indirect effects. The first step in constructing a structural equation model is to determine candidate mediator variables to be included in the model. To do so, I first investigated how best to represent the knowledge students acquire as they are working on the tutor problems by comparing different knowledge component models. Second, I used the knowledge component model to generate a number of features that describe student performance during problem solving. Third, I searched for features that are predictive of learning outcome, using linear regressions.

5.1 Knowledge component model

First, I constructed a knowledge component model that adequately describes knowledge students acquire when working with interactive representations to learn about atomic structure. Knowledge components are "acquired units of cognitive function or structure that can be inferred from performance on a set of related tasks" [19]. I contrasted the following knowledge component models:

1. A *single-step baseline model* that treats all problem-solving step as one skill;
2. A *step-type model* that does not distinguish between the graphical representation used in the given problem but distinguishes between step types (i.e., providing information about atoms, planning the graphical representation of the atom, constructing graphical representations, and making inferences about the atom; see Figures 2 and 3);
3. A *representation-construct model* that distinguishes between the graphical representation used in the given problem (i.e., Lewis structure, Bohr model, energy diagram, and orbital diagram; see Figure 1) for the step in which students are asked to construct the graphical representation, but that does not distinguish between graphical representations for the remaining step types;
4. A *step-type / representation model* that distinguishes between the graphical representation used in the given problem for each step types except for providing information about atoms.

Each model was evaluated as to how well it predicts student behavior during problem solving. Following standard practice in ITS research [19, 20], I considered each step in a given tutor problem as a learning opportunity for the particular knowledge component involved in the step. Student behavior was assessed based on whether a student solved the step correctly (i.e., without hints and without errors). To evaluate model fit, I used the Additive Factors Model (AFM) in the PSLC DataShop [20]. As a metric for model fit, I used 3-fold item-stratified cross validation [21]. Table 1 shows the root mean squared errors (RMSEs) for each knowledge component model. The *step-type / representation* model had the best model fit. Hence, this knowledge component model was used as a basis to generate features that describe students' learning about atomic structure with interactive graphical representations.

Table 1. RMSEs for knowledge component models.

Knowledge component model	Knowledge components	Item-stratified RMSE (lower is better)
Single-step baseline model	1	0.464794
Step-type model	4	0.375733
Representation-construct model	7	0.372553
Step-type / representation model	13	0.363908

5.2 Feature generation

Based on the step-type / representation model, I generated features that describe how students interact with the tutor problems. Students' problem-solving behaviors can be described based on the outcome (proportion of incorrect first attempts, proportion of hint requests at the first attempt, proportion of total incorrect attempts, proportion of total hint requests) and based on durations (time spent per step in total, time spent on steps with first correct attempt / steps with at least one incorrect attempt, time spent before first attempt, time spent before first attempt if it was a correct / incorrect attempt). Additionally, when students use an interactive tool (e.g., to construct representations) they can make a large variety of errors. Thus, the number of different error types when constructing representations is another measure of interest. To generate features, I computed these metrics for each knowledge component, yielding a total of 134 features (i.e., four outcome-based and six duration-based set of metrics for each of the 13 KCs, plus number of mistake types for constructing each of the four representations).

5.3 Search for predictive features

Since it is impractical to include all 134 features in a structural equation model, it was necessary to narrow down the number of features to consider. The most interesting features when investigating the role of spatial skills on learning outcomes are those features that are predictive of students' learning outcomes. To find predictive features, I conducted linear regressions on each set of features (i.e., proportion of correct steps, time spent on correct steps, etc.), computed for the given KCs. It was necessary to conduct separate regressions for each set of feature because the feature sets are not independent of one another. For example, the total incorrect attempts subsume the first incorrect attempts. Learning outcomes on the final posttest was the dependent variable in each linear regression model. Pretest performance was included as a predictor in all regression models. Regressions were conducted using 10-fold cross-validation. I used the results from the regression analyses to determine what characterizes predictive features. To do so, I compared the standardized coefficients and significance of features based on the metric they used and based on the KC they described. Table 2 shows the results for the regression analyses.

The goal of the selection procedure was to identify a set of predictive features that are independent of one another. Overall, features based on *knowledge components* related to planning, constructing, and making inferences were predictive of learning outcomes. However, features based on retrieving information about atoms were not predictive of learning outcomes. Thus, atoms steps were excluded from further analysis. Among the *outcome-based features*, those using proportion of incorrect first attempts and those using proportion of total incorrect attempts were equally predictive of learning outcomes. However, when excluding atoms steps, the features based on proportion of incorrect total attempts were slightly more predictive than those based on incorrect first attempts. Thus, features based in incorrect total attempts were selected for further analysis. Features based on proportion of hint requests at first attempt and proportion of total hint requests had low predictive value because hint use was generally low. Thus, these features were excluded. Features describing error types while constructing representations had high predictive value. Thus, these features were selected for further analysis. Among the *duration-based features*, those based on time spent on steps with at least one incorrect attempt as a metric were selected because they were more predictive than the other duration-based features.

Based on these findings, the following variables were selected for the structural equation model:

- Average duration of planning steps with at least one incorrect attempt (plan_timeError)
- Average duration of representation-construction steps with at least one incorrect attempt (repr_timeError)
- Average duration of inference steps with at least one incorrect attempt (infer_timeError)
- Proportion of total incorrect attempts on planning steps (plan_incorrect)
- Proportion of total incorrect attempts on representation-construction steps (repr_incorrect)
- Proportion of total incorrect attempts on inference steps (infer_incorrect)
- Number of error types on representation-construction steps (repr_errorTypes)

Table 2. Standardized coefficients for mediators in regression models, using color gradients to illustrate the strength of association with performance on the final posttest.

predictor	outcome-based features			duration-based features					
	total incorrects	incorrect 1st attempt	error-Types	total step duration	correct step duration	error step duration	before 1st attempt	before 1st correct	before 1st error
pretest	0.275	0.281	0.307	0.364	0.356	0.258	0.334	0.372	0.305
atom	-0.002	-0.027		0.013	0.009	-0.076	0.006	-0.007	-0.054
planning-Bohr	0.112	0.082		-0.137	0.018	-0.039	0.068	0.024	0.016
planning-Energy	-0.393	-0.112		-0.163	-0.001	0.230	0.075	0.036	0.025
planning-Lewis	-0.116	-0.114		-0.025	-0.006	-0.048	-0.118	-0.093	-0.046
planning-Orbital	0.018	0.112		-0.004	0.112	-0.118	-0.066	0.07	-0.071
construct-Bohr	-0.028	0.230	-0.201	-0.080	-0.053	-0.050	0.031	-0.103	0.062
construct-Energy	-0.030	-0.174	-0.093	0.269	-0.144	0.003	0.087	-0.086	-0.155
construct-Lewis	0.203	-0.053	-0.169	-0.109	0.025	-0.113	-0.077	0.029	-0.158
construct-Orbital	-0.028	-0.119	0.139	0.056	0.045	-0.166	-0.211	0.064	-0.202
inference-Bohr	-0.030	0.011		-0.080	-0.138	-0.114	-0.017	-0.046	0.059
inference-Energy	-0.121	-0.064		0.269	0.196	0.091	0.121	0.064	0.116
inference-Lewis	0.071	0.040		0.169	0.013	-0.093	-0.023	0.025	0.053
inference-Orbital	-0.140	-0.147		-0.107	-0.044	-0.106	0.075	0.039	0.010
Average of absolute values	0.112	0.112	0.182	0.132	0.083	0.108	0.094	0.076	0.095

6. STRUCTURAL EQUATION MODEL

The goal of the structural equation model was to investigate why students with low spatial skills show lower learning gains. The structural equation model allows testing whether students' problem-solving behaviors statistically mediate the effect of spatial skills on learning gains. To carry out this analysis, I considered the variables that I identified as predictive of students' learning outcomes as potential mediators of the effect of spatial skills on learning outcomes at the final posttest, controlling for pretest.

6.1 Model Search

Since there are many models that might describe the nature of the effect of spatial skills on learning outcomes, I conducted a model search. Because a factor analysis indicated that the chemistry content pretest and the mental rotation ability test load onto separate factors that correlate weakly, I assumed that pretest and spatial skills are independent. I assumed that pretest is prior to the mediators and to the final posttest, that spatial skills are prior to the mediators and to the final posttest, and that mediators are prior to the final posttest. For the mediators, I assumed that planning is prior to constructing representations, which is prior to making inferences. Even under these constraints, there are at least 2^{49} distinct models that are consistent with these assumptions. Figure 4 shows the fully saturated model that would be compatible with these assumptions. A fully saturated model contains all possible edges (or "effects") compatible with the assumptions. Therefore, Figure 4 illustrates the search space of models: the search was conducted among models that had all, none, or a subset of the edges in the fully saturated model.

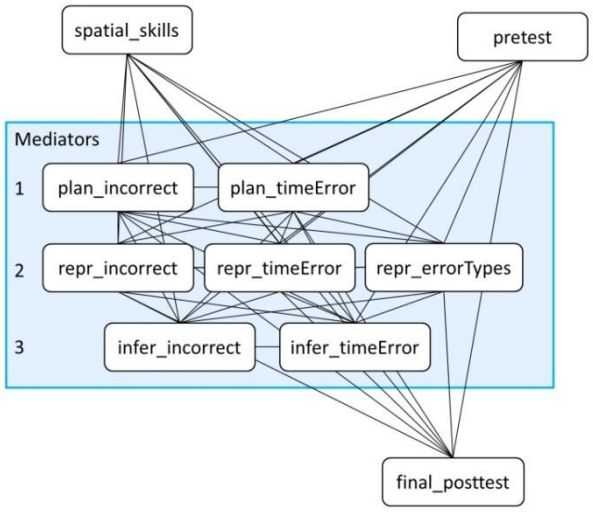


Figure 2. Fully saturated model consistent with the assumptions. Mediators are highlighted in blue and organized by tiers (1 = planning; 2 = representation-construction, 3 = inference).

To search for models that are theoretically plausible and consistent with the data, I used the Tetrad V program's[1] GES algorithm along with background knowledge constraining the space of models searched [22] to those that are theoretically tenable and compatible with my assumptions [23]. In the model search, each edge shown in Figure 4 is evaluated as to whether including it yields a better model fit than not, and whether it is a statistically reliable effect. As Figure 4 illustrates, there are many distinct models consistent with the background knowledge and that are plausible tests for the mediation hypothesis. Yet, it is important to know which of these models fits the data best, because parameter estimates and the statistical inferences we make about them are conditional on the model being true. Parameter estimates of models that do not fit the data well are scientifically unreliable. Thus, searching for the model that is most consistent with the data ensures that the parameters of the model can be trusted.

To conduct the model search at a technical level, I represented the qualitative causal structure of each model by a Directed Acyclic Graph (DAG). If two DAGs entail the same set of constraints on the observed covariance matrix,[2] then they are empirically indistinguishable. If the constraints considered are independence and conditional independence, which exhaust the constraints entailed by DAGs among multivariate normal varieties, then the equivalence class is called a *pattern* [23, 24]. The GES algorithm is asymptotically reliable,[3] and outputs the *pattern* with the best BIC score.[4] The pattern identifies features of the causal structure that are distinguishable from the data and background knowledge, as well as those that are not. The algorithm's limits lie primarily in its background assumptions involving the non-existence of unmeasured common causes and the parametric assumption that causal dependencies can be modeled with linear functions. The outcome of the model search is a structural equation model model that (1) is theoretically plausible, (2) fits the data well, and (3) contains only edges that describe statistically reliable effects.

6.2 Results

Figure 5 shows a model found by GES, with unstandardized parameter estimates. Table 2 shows standardized parameter estimates. Each edge is evaluated as to whether it is a reliable effect using *t*-tests, assuming an alpha-level of .05. A Bonferroni correction of the *p*-values is not necessary in a structural equation model because the significance tests are not independent. Table 2 shows the results from these tests. Altogether, the model fits the data well[5] (χ^2 = 32.77, df = 27, p = .21).

Table 3. Parameter estimates (PE) for all edges and result of *t*-tests assessing whether the PE is significantly different from 0.

Edge from…	to…	PE	*t*	*p*
infer_timeError	infer_incorrect	.0124	3.2999	.0013
plan_incorrect	final_posttest	-.1116	-2.4706	.0150
plan_incorrect	infer_incorrect	.3704	6.3202	< .001
plan_incorrect	plan_timeError	4.6959	3.7759	< .001
plan_incorrect	repr_incorrect	3.0573	9.5622	< .001
plan_incorrect	repr_timeError	19.8158	2.8253	.0056
plan_timeError	infer_incorrect	-.0079	-2.2244	.0281
plan_timeError	infer_timeError	.2706	3.1104	.0024
plan_timeError	repr_timeError	1.8936	4.7591	< .001
pretest_content	final_posttest	0.2293	2.9336	.0040
pretest_content	plan_incorrect	-.4975	-3.1908	.0018
repr_incorrect	infer_incorrect	.0329	2.6633	.0088
repr_incorrect	repr_timeError	11.068	7.529	< .001
repr_timeError	infer_timeError	.0394	3.1303	.0022
repr_timeError	repr_errorTypes	.0083	8.6891	< .001
spatial_skills	final_posttest	.147	1.9078	.0589
spatial_skills	plan_incorrect	-.4102	-2.6326	.0096
spatial_skills	plan_timeError	-3.5242	-1.639	.1039

The final model shows that spatial skills have a direct positive effect on students' learning outcomes at the final posttest. Furthermore, spatial skills predict students' problem-solving behaviors while they are planning the graphical representation, which, in turn, has an effect on outcome-based and duration-based measures of problem-solving behaviors while they construct the graphical representation and while they make inferences from graphical representations about domain-relevant concepts. Only the proportion of incorrect attempts on planning steps mediates the effect of spatial skills on learning outcomes: plan_incorrect is the only variable that mediates the effect of spatial_skills on final_posttest. The edge from spatial_skills to plan_incorrect shows that a student with a perfect score on the spatial skills test makes .4102 fewer incorrect attempts per step than a student with the lowest possible score on the spatial skills test. The edge from plan_incorrect to final_posttest means that a student who makes one incorrect attempt per step scores 11.16% lower on the final posttest than a student who makes no incorrect attempts (controlling for pretest performance). In sum, the mediated effect of spatial_skills to final_posttest through plan_incorrect is .4102 * .1116 = .0458. Incorrect attempts while planning representations only partially mediate the effect of spatial skills on learning outcomes, because there is a direct effect of .147 from spatial_skills to final_posttest. Yet, making more incorrect attempts while planning graphical representations explains a considerable portion (about 25%) of the effect of spatial skills on learning outcomes.

7. CONCLUSIONS

The goal of the mediation analysis was to investigate (1) which aspects about working with interactive representations are harder for students with low than with high spatial skills and (2) which of these aspects explain why students with low spatial skills show lower learning gains than students with high spatial skills. With respect to the first question, results show that spatial skills have an effect on all aspects of students' problem-solving behaviors,

[1] Tetrad, freely available at www.phil.cmu.edu/projects/tetrad, contains a causal model simulator, estimator, and over 20 model search algorithms, many of which are described and proved asymptotically reliable in [24].

[2] An example of a testable constraint is a vanishing partial correlation, e.g., $\rho_{XY.Z} = 0$.

[3] Provided the generating model satisfies the parametric assumptions of the algorithm, the probability that the output equivalence class contains the generating model converges to 1 in the limit as the data grows without bound. In simulation studies, the algorithm is quite accurate on small to moderate samples.

[4] All the DAGs represented by a pattern will have the same BIC score, so a pattern's BIC score is computed by taking an arbitrary DAG in its class and computing its BIC score.

[5] The usual logic of hypothesis testing is inverted in path analysis: a *low* p-value means the model can be rejected.

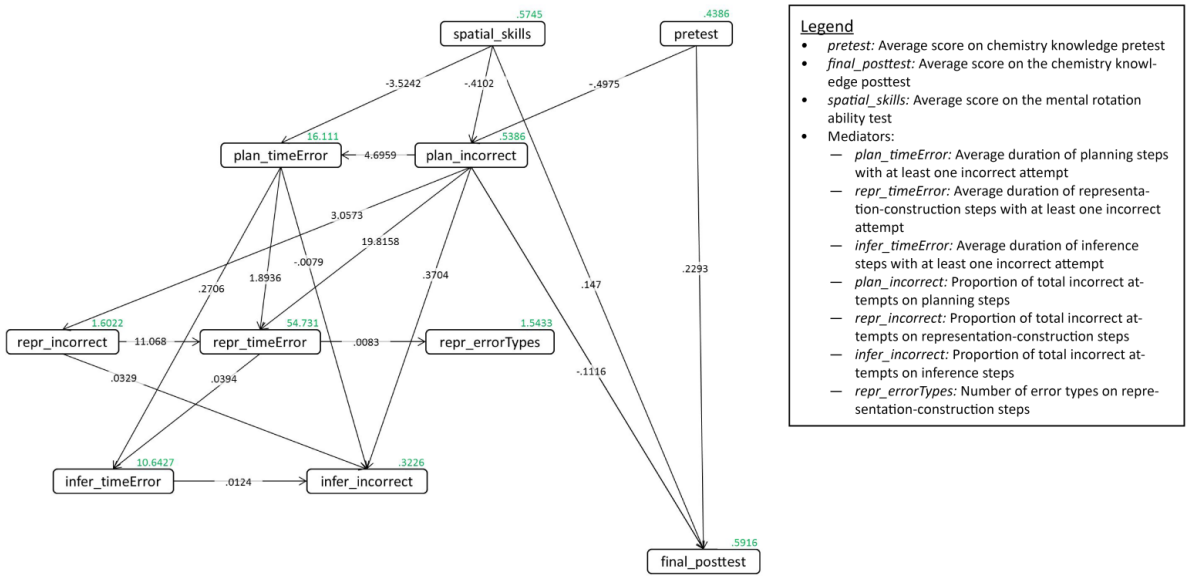


Figure 3. Final structural equation model with unstandardized parameter estimates. Green values show means.

except for looking up information about the atoms: planning, constructing, and making inferences from graphical representations. Spatial skills affect outcome-based measures of performance as well as duration-based measures of performance. Yet, the structural equation model shows that planning has a central role: students' ability to plan graphical representations has an impact on all further problem-solving behaviors as students construct graphical representations and make inferences about domain-relevant concepts based on the graphical information. With respect to the second question, results show that planning is the only aspect that mediates the effect of spatial skills on learning gains. The difficulties that students with low spatial skills have in constructing representations and in making inferences may merely be symptomatic—they do not explain why these students show lower learning gains. Only the fact that students with low spatial skills tend to struggle more in planning representations explains why they benefit less from interactive representations.

Why might students' ability to plan graphical representations be so strongly affected by their spatial skills? Planning a representation requires students to describe what the representation should look like, based on the properties of the atom. This task requires them to mentally picture visuo-spatial features based on text-based information about the atom's properties. This takes more cognitive effort for students who struggle with such visuo-spatial tasks. Hence, these students are at risk of cognitive overload during planning, which jeopardizes learning. Perhaps difficulties in planning are amplified by the fact that the interactive representation tool is not visible during the planning step (see Figure 3).

Why might the ability to plan representations determine students' learning gains? Learning with graphical representations means that students have to visualize new information externally while integrating this information with their internal mental models of the domain-relevant concepts [26]. Planning might play a central role because it helps students organize their initial mental model of the domain-relevant concepts. Having a well-organized initial mental model might facilitate integration of new information into this model: learning occurs as students expand and repair their mental models throughout the learning intervention, for instance by self-explaining how the new information relates to their initial mental models [27].

In summary, the findings from the mediation analysis shed light into the broader theoretical question of how spatial ability affects learning outcomes in STEM. Spatial skills seem to be important because students' benefit from interactive representations depends on their ability to mentally visualize abstract concepts *before* they use an external representation to visualize the concept. Mental visualization may play a key role in students' learning of abstract concepts because it allows students to integrate new information into their mental models. These findings also yield new hypotheses about the practical question of how best to support students with low spatial skills. These students might benefit from receiving additional assistance in planning graphical representations. They might benefit from seeing the interactive representation tool during the planning steps, so that they can more easily visualize the representation. They may also benefit from receiving examples of successful planning. It would be interesting to investigate whether such support increases learning gains for students with low spatial skills. In light of the interpretation that planning is so important because it helps students organize their initial mental models, it would be interesting to conduct a think-aloud study to assess whether, indeed, helping students plan representations facilitates mental model integration.

Several limitations of the present analysis need to be discussed. First, performance on planning steps only partially mediates the effect of spatial skills on learning outcomes. Thus, there might be other mediators that we did not assess. Further research is needed to investigate other aspects of problem solving that explain why students with low spatial skills tend to show lower learning gains. Second, the data is correlational: it is impossible to randomly assign students to having "low" or "high" spatial skills. As in any correlational data set, there may be other unknown factors that affect the effects of interest. Third, the structural equation model assumes linear relations between the variables in the model. This assumption is reasonable but not infallible. Finally, the analysis is based on a sample of 117 students. Even though that is sizable compared to many ITS studies, model search reliability increases with sample size, but decreases with model complexity. Hence, it is impossible to put confidence bounds on finite samples [21].

To conclude, the mediation analysis presented in this paper yields new insights into why students with lower spatial skills struggle in

learning with interactive graphical representations. It seems that planning representations is a crucial aspect of learning success. This finding yields new hypotheses about what types of interventions these students may benefit from. Even though the present paper merely presents a first step towards better understanding the mechanisms that underlie the "the-rich-get-richer" rule in STEM domains, it may help us address the unfortunate fact that students with low spatial skills tend to show lower achievements in STEM domains and they are less likely to pursue careers in these domains. In other words, this paper is a first step towards creating an "everyone-gets-richer" rule for STEM learning.

8. ACKNOWLEDGMENTS

This work was supported by the UW-Madison Graduate School and WCER. We thank Teri Larson, Ned Sibert, Stephen Block, Amanda Evenstone, and Jocelyn Kuhn for their help with recruitment, and Sally Wu and the RAs in the Learning, Representations, & Technology Lab for their help in conducting the experiment.

9. REFERENCES

[1] Uttal, D.H., Meadow, N.G., Tipton, E., Hand, L.L., Alden, A.R., Warren, C., Newcombe, N.S.: The malleability of spatial skills: a meta-analysis of training studies. Psychological Bulletin 139, 352-402 (2013)

[2] Wai, J., Lubinski, D., Benbow, C.P.: Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. Journal of Educational Psychology 101, 817-835 (2009)

[3] Barnea, N., Dori, Y.J.: High-school chemistry students' performance and gender differences in a computerized molecular modeling learning environment. Journal of Science Education and Technology 8, 257-271 (1999)

[4] Stieff, M.: Sex differences in the mental rotation of chemistry representations. Journal of Chemical Education 90, 165-170 (2013)

[5] Stieff, M.: Mental rotation and diagrammatic reasoning in science. Learning and Instruction 17, 219-234 (2007)

[6] Kozma, R., Russell, J.: Students becoming chemists: Developing representationl competence. In: Gilbert, J. (ed.) Visualization in science education, pp. 121-145. Springer, Dordrecht, Netherlands (2005)

[7] Stieff, M., Hegarty, M., Deslongchamps, G.: Identifying representational competence with multi-representational displays. Cognition and Instruction 29, 123-145 (2011)

[8] Griffiths, A.K., Preston, K.R.: Grade-12 students' misconceptions relating to fundamental characteristics of atoms and molecules. Journal of Research in Science Teaching 29, 611-628 (1992)

[9] Hegarty, M., Waller, D.A.: Individual differences in spatial abilities. In: Shah, P., Miyake, A. (eds.) The Cambridge handbook of visuospatial thinking, pp. 121–169. Cambridge University Press, New York, NY (2005)

[10] Clements, D.H.: 'Concrete' Manipulatives, Concrete Ideas. Contemporary Issues in Early Childhood 1, 45-60 (1999)

[11] Ai-Lim Lee, E., Wong, K.W.: Learning with desktop virtual reality: Low spatial ability learners are more positively affected. Computers & Education ahead of print (2014)

[12] VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. Educational Psychologist 46, 197-221 (2011)

[13] Tuckey, H., Selvaratnam, M., Bradley, J.: Identification and rectification of student difficulties concerning three-dimensional structures, rotation, and reflection. Journal of Chemical Education 68, 460-464 (1991)

[14] Davidowitz, B., Chittleborough, G.: Linking the macroscopic and sub-microscopic levels: Diagrams. In: Gilbert, J.K., Treagust, D.F. (eds.) Multiple representations in chemical education, pp. 169-191. Springer, Dordrecht, Netherlands (2009)

[15] Rau, M.A., Michaelis, J.E., Fay, N.: Connection making between multiple graphical representations: A multi-methods approach for domain-specific grounding of an intelligent tutoring system for chemistry. Computers and Education 82, (2015)

[16] Ainsworth, S.: DeFT: A conceptual framework for considering learning with multiple representations. Learning and Instruction 16, 183-198 (2006)

[17] Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. International Journal of Artificial Intelligence in Education 19, 105-154 (2009)

[18] Rau, M.A., Wu, S.P.W.: ITS support for conceptual and perceptual processes in learning with multiple graphical representations. submitted to AIED 2015 (under review)

[19] Peters, M., Laeng, B., Latham, K., Jackson, M., Zaiyouna, R., Richardson, C.: A Redrawn Vandenberg & Kuse Mental Rotations Test: Different Versions and Factors that affect Performance. Brain and Cognition 28, 39-58 (1995)

[20] Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Cognitive Science 36, 757–798 (2012)

[21] Koedinger, K.R., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: The PSLC Data-Shop. In: Romero, C. (ed.) Handbook of educational data mining, pp. 10-12. CRC Press, Boca Raton, FL (2010)

[22] Stamper, J.C., Koedinger, K.R., McLaughlin, E.A.: A Comparison of Model Selection Metrics in DataShop. In: D'Mello, S.K., Calvo, R.A., Olney, A. (eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013), pp. 284-287. International Educational Data Mining Society (2013)

[23] Chickering, D.M.: Optimal Structure Identification with Greedy Search. Journal of Machine Learning Research 3, 507-554 (2002)

[24] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT Press (2000)

[25] Pearl, J.: Causality: Models, Reasoning, and Interference. Cambridge University Press (2000)

[26] Bodner, G.M., Domin, D.S.: Mental models: The role of representations in problem solving in chemistry. University Chemistry Education 4, 24-30 (2000)

[27] Wylie, R. and Chi, M.T., 2014. The Self-Explanation Principle in Multimedia Learning. In *The Cambridge Handbook of Multimedia Learning*, R.E. Mayer Ed. Cambridge University Press, New York, NY, 413-43

SHORT PAPERS

Spectral Bayesian Knowledge Tracing

Mohammad Falakmasir
University of Pittsburgh
210 South Bouquet Street,
Pittsburgh, PA 15213
(412) 624-5755
falakmasir@pitt.edu

Michael Yudelson
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219
(412) 690-2442
myudelson@
carnegielearning.com

Steve Ritter
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219
(412)-690-2442
sritter@
carnegielearning.com

Ken Koedinger
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
(412)-268-7667
koedinger@cmu.edu

ABSTRACT
Bayesian Knowledge Tracing (BKT) has been in wide use for modeling student skill acquisition in Intelligent Tutoring Systems (ITS). BKT tracks and updates student's latent mastery of a skill as a probability distribution of a binary variable. BKT does so by accounting for observed student successes in applying the skill correctly, where success is also treated as a binary variable. While the BKT served the ITS community well, representing both the latent state and the observed performance as binary variables is, nevertheless, a simplification. In addition, BKT as a two-state and two-observation first-order HMM is prone to noise in the data. In this paper, we present work that uses feature compensation and model compensation paradigms in an attempt to conceptualize a more flexible and robust BKT model. Validation of this approach on the KDD Cup 2010 data shows a tangible boost in model accuracy well over the improvements reported in the literature.

Keywords
Cognitive model of student practice, Bayesian Knowledge Tracing.

1. INTRODUCTION
Bayesian Knowledge Tracing (BKT) is one of the most popular student modeling techniques in the field of Intelligent Tutoring Systems (ITS). It has been used for 20 years now, and it has served the educational community well. Among the major weaknesses of BKT are the non-identifiability of the parameters, parameter degeneracy [1], and, in general, susceptibility to the noise in the naturally-occurring data. BKT is, by definition, a first-order Hidden Markov Model (HMM) with a binary latent variable representing student knowledge and a binary observed variable indicating student performance. While representing latent student knowledge as a binary variable with *known* and *unknown* states has been widely accepted by the Intelligent Tutoring Community (ITS), it is, no doubt, a simplification. Accounts of the need for a larger number of latent states can be found in the literature, including but not limited to the work of Aleven et al. [2].

Practical issues occur in other fields where first-order HMMs are used intensively (e.g., speech recognition, handwriting recognition, etc.). In these fields, it is common to adopt various compensation measures including model compensation and feature compensation [3]. In this paper, we are applying both compensation paradigms to create a variant of BKT – Spectral BKT – in an attempt to overcome some of BKT's shortcomings. Spectral BKT uses spectral observations – *n*-grams of the consecutive original unary observations of correct and incorrect skill application. It also relies on an extended set of latent states. While a number of Spectral BKT configurations can be conceived, we constructed and empirically tested a setup with eight spectral observations (3-grams of original observations) and four states. To validate the Spectral BKT approach uses an openly available KDD Cup 2010 data set of the 2008-2009 Carnegie Learning's Cognitive Tutor data. The resulting improvement is well above all reported in the literature.

2. RELATED WORK
2.1 Bayesian Knowledge Tracing
Bayesian Knowledge Tracing (BKT) was introduced by Corbett and Anderson [4] in 1995. The standard BKT model assumes that student knowledge of a particular skill is an unobserved binary latent variable that changes based on the binary correctness of the observed student performance. Standard BKT has 4 parameters. Probability of knowing skill a priori (pInit), probability of learning the skill after each opportunity to apply it (pLearn), probability of making a mistake when applying an already known skill (pSlip), and probability of luckily producing a correct response when the skill is not known (pGuess). The probability of knowledge decay (pForget) is assumed to be zero in standard BKT. In general, a HMM with two states and two observations that has a total of 10 numeric values would be said to have 5 parameters (last value in every row is redundant). However, since forgetting is set to zero, BKT is assumed to have 4 parameters.

A large volume of work has been published on fitting BKT models and its variations. Wang and Beck [6] introduced two hierarchical factors into BKT to account for and compare class and student level parameter variability. Xu and Mostow [7] blend BKT approach with logistic regression and create an LR-DBN model that is capable of addressing multiple skill coding for a single step (something that BKT technically doesn't, due to conditional independence assumptions). González-Brenes et al. [8] generalized BKT model to address a feature-rich context addressing multiple skills per step, temporal features, and expert knowledge. Another work of Pardos and Heffernan is an extension of BKT call KT-IDEM [9]. It addressed item variance in the data via introducing item difficulty observable nodes.

2.2 Empirical Problems of BKT
A noticeable portion of the work on BKT models is devoted to discussing problems researchers face when fitting them to the data. Baker et al. [1], when talking about the contextual estimation of guess and slip parameters in BKT, stipulate that their model is less prone to the BKT model degeneracy. What is often meant by

degeneracy are the cases when probabilities of slipping and guessing assume unjustifiably high values, and this often calls for the use of parameter caps. BKT model degeneracy is the artifact of the known issue in HMM called label switching [10]. The issue is made more convoluted by the fact that forgetting is not allowed to vary in BKT and is set to zero.

Work by Beck and Chang [11] discusses an example of yet another problem of BKT – identifiability. There often exists a range of parameter value sets that result in the same likelihood given the data it's estimated on. Falakmasir and colleagues [12] have encountered the same problem in their previous work on the Spectral Learning approach to fitting BKT models. In that work, the formulation of the best-fitting parameter search problem was transformed into the spectral space, where a global optimum of the objective function is guaranteed to be reached. When translating the spectral solution back to the HMM space, the authors had to define a heuristic to pick the most *plausible* parameters from an infinite set of equally good parameter sets.

2.3 Theoretical Issues with BKT

Arguably, it's the two Markov assumptions and the setup of the BKT that result in its known shortcomings. First, is the Limited Horizon Assumption states that the probability of being in a state at time *t* depends only on the state at time *t*-1. This kind of HMM is called a first-order HMM since it only has a *memory* of one previous time slice. Second Markovian assumption is the Stationary Process Assumption that the conditional distribution over the next state given the current state does not change over time. Given the fact that BKT has only one parameter to capture state transition, student learning rate is forced to remain constant.

Both, the limited *memory*, and the constant learning rate are simplifications and one can easily construct a case for a more flexible representation of skill learning. For example, between the *unknown* state and *known* state there can be states that capture the preliminary stage of learning when the student having just seen one or two problems is mostly guessing. Before transitioning to the known state, the skill could be in the state that often results in slips since student's knowledge is not strong enough. Another likely reason for BKT's limitations is sensitivity to noise. In BKT, Gaussian noise is assumed for the latent (knowing the skill) and the observed variables. However, when dealing with naturally occurring data, the signal to noise ratio might drop considerably. As a result, one might arrive at degenerate model parameters.

There are two main approaches to handling noise in HMM: feature compensation and model compensation. In feature compensation, the noisy traits (for example, observations) are enhanced to remove the effect of the noise. In model compensation, the original models are mapped into a new model that can be learned from the noisy observations. It has been empirically established that feature compensation is simpler and more efficient to implement, but model compensation has the potential for the greater robustness [3].

3. SPECTRAL BKT

In this work, we are attempting to combine feature compensation and model compensation to overcome the shortcomings of the standard BKT that assumes an ideal noise-free environment and is represented by a first-order HMM. We address feature compensation by changing the way we treat the observations. Instead of a single observation, we are considering *n*-grams – sequences of consecutive observations for the skill, where next *n*-gram observation inherits *n*-1 atomic observations from the previous one. In NLP, 3-grams are often successfully used for feature compensation and we have empirically found that 3-grams work sufficiently well while 2-grams do not. From the information-theoretic point of view, the entropy rate of Hidden Markov Processes with two states proved to have at most second order behavior (captured by second-order HMM) [13]. This means that if we consider the data to be generated by a relatively noise-free naturally-occurring process and that the skills are fine-grained enough, we only need to look at 3-grams of the observations in order to find the true model. One may use *n*-grams with *n* greater than 3. However, the computations involved would grow exponentially. Figure 1 shows how the original sequence of observations is encoded into 3-grams.

The model compensation is addressed by adding two intermediate states between the *unknown* and *known* to the original BKT. Once the new observations are defined, the new model that we will call Spectral BKT (due to the use of spectral observations) can be treated as a first order HMM for the purposes of fitting the parameters.

In Spectral BKT, state 1 is the *known* state and state 4 is the *unknown* state. States 2 and 3 we leave unlabeled at this point. Like in the standard BKT, once the student is in the *known* state we assume no un-learning. Moreover, the probability of going from *the unknown* state directly to *the known* state is zero. Finally, once the knowledge transitions from the *unknown* state, there's no return. Given these assumptions, the sparsity structure changes the number of state transition parameters from 1 in standard BKT to 6 in Spectral BKT. By enforcing the sparsity structure in our transition matrix, we guarantee the forward progressing from unknown to known in each iteration and prevent the EM algorithm from learning degenerate models. We assume no further sparsity in any of the 4 priors and 4*7=28 values of the observation matrix, we have (4-1)+6+(7-2)*4=37 parameters in this particular Spectral BKT conceptualization.

The transformation of the original data for fitting the new Spectral BKT is fairly simple (rf. Figure 1). However, when we talk about model predictions, the Spectral BKT produces probability distributions over 8 3-gram observations and one has to make special arrangements to convert them to 2 (probability of correct and of incorrect) in order to compare it with the standard BKT algorithm fairly. First, we ordered the spectral observations from 000 to 111 linearizing a partial order heuristic (rf. Figure 2a). According to this heuristic a spectral observation 011 is the second best indication of success after observation 111. Spectral observation 101 is third best with, potentially, a careless slip in the middle. Spectral observations 001 and 110 were a judgment call. We have placed 001 before 110, assuming it is an early indicator of learning, and 110 is a premature indicator of learning with a failure in the end.

When mapping 8 values to binary success and failure, we came up with three rules. A *regular* rule splits 8 probabilities exactly in half and sums of the two groups are the new probability of correct and incorrect (third column in Figure 2a). The *regular* rule can also be interpreted as looking at the third bit of each 3-gram A *strict* rule is more stringent about which observation probabilities are counted toward success. A *relaxed* rule is more. Since our Spectral BKT produces a first 8-probability predictions starting with the third original observation (due to the use of 3-grams), we have also devised mapping of the 8 probabilities to produce predictions for the first two observations. These mappings are given in Figure 2b,c and reference the spectral observations from Figure 2a. For example, if the observed data contained observations 0, 1, and 0, and the Spectral BKT prediction of

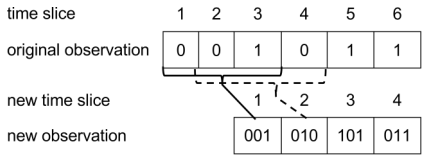


Figure 1. New *n*-gram observations

	Spectral obs.	Regular	Strict	Relaxed
1	111	1	1	1
2	011	1	1	1
3	101	1	1	0
4	001	1	1	0
5	110	0	1	0
6	100	0	0	0
7	010	0	0	0
8	000	0	0	0

(a)

	Spectral obs.	Regular	Strict	Relaxed
1,5	11*	1	1	1
2,7	01*	1	0	1
3,6	10*	0	0	1
4,8	00*	0	0	0

(b)

	Spectral obs.	Regular
1,3,5,6	1**	1
2,4,7,8	0**	0

(c)

Figure 2. Mapping spectral observations from a distribution over 8 probabilities to 2: a) predicting starting with a 3rd original observation when two prior observations are available. b) & c) predicting original observations 2 and 1.

correctness was {0, 0, 0.1, 0.1, 0.1, 0.2, 0.2, 0.3}, then, according to the regular mapping rules, probabilities of correct for the three observations would have been 0.4, 0.3, and 0.2.

4. DATA

To validate our models we used data from KDD Cup 2010 donated by Carnegie Learning, Inc. and available for downloading at http://pslcdatashop.web.cmu.edu/KDDCup. Of the two datasets available we chose Bridge to Algebra. This dataset contains about 20 million transactions belonging to over 6 thousand students working on nearly 150 sections of mathematics curriculum practicing around 1650 skills. The dataset contains information about curriculum context (unit and section the student is in), problem context (problem name and problem step name), cognitive skill labels, timing, as well as correctness of the first attempt to solve the problem step and assistance information (number of hints requested and number of errors). The KDD Cup 2010 is currently the largest freely available collection of learner data. That and the fact that this data was collected by Carnegie Learning's Cognitive Tutor that uses BKT model makes it a good candidate for testing the Spectral BKT. According to the custom of the Carnegie Learning's Cognitive Tutor, skills were considered unique within each curriculum section even if the skill label repeated across several sections. Also, we have treated an absence of the skill (a null skill) as a special skill.

5. MODEL VALIDATION

For the purposes of training the models, we have transformed the original data with unigram observations into a dataset with 3-gram observations. We ran 10-fold student-based and item-based cross-validations that each produced a set of predictions for the transformed 3-gram data. To fit and cross-validate the models we used the `hmmsclbl` tool – a C/C++ utility specially developed to work with large data sets and successfully used in [5] (available for download at http://github.com/IEDMS/standard-bkt). Standard BKT outputs two predictions per data row – probability of correct application of the skills in question and the probability of incorrect application. Spectral BKT works with 8 spectral observations and its predictions come in the form of probability distributions of 8 values per row of the predicted data. Spectral BKT models predictions were mapped from the 8-values onto the 2-value probability distribution schema in Figure 2. The summary of the cross-validation results for the training dataset is listed in Table 1. Here we list the performance of standard BKT next to the performance of Spectral BKT model. We only list results the *relaxed* 3-gram-to-unigram mapping, since *regular* and *strict* mapping performed worse. We tested several solver algorithms `hmmsclbl` supports, including EM and stochastic gradient descent. EM gave a consistently better performance, but the margin was small: within 1% in accuracy and 0.03 in RMSE.

When running student-stratified cross-validation, we were repeatedly *hiding* the full data belonging to 10% of the students. In item-stratified cross-validation, the transactions belonging to problems that we intended to *hide* could appear in individual students' data in arbitrary locations. For the purposes of item-stratification, we have marked the data of 10% of the items as unobserved but accounted for the opportunity to apply skills.

Standard BKT model has 4 parameters per skill. Spectral BKT model, as per our conceptualization of the transition matrix, has 37. The number of parameters being an order of magnitude higher, the AIC and BIC metrics that penalize for that go up 3% and 9% (item-stratified cross-validation). In the case of student-stratified cross-validation, both AIC and BIC are decreaseв by 21% and 13%. Accuracy and RMSE in case of Spectral BKT improve a lot. To the best of our knowledge, the overall accuracy of BKT or its variations was never reported to be above 90% on the dataset we used and Spectral BKT hits an impressive 92%.

Recall that we had to *back-predict* the predictions of Spectral BKT for student skill opportunities one and two due to the use of 3-gram observations. For this purpose, in Table 1 we list the additional accuracy and the RMSE values for student skill opportunity 1 alone (7% of the data), opportunity 2 alone (6% of the data), and opportunity 3 and further (87% of the data). To no surprise, the first opportunity prediction of Spectral BKT is slightly worse than the one of standard BKT by a margin in the

Table 1. Comparison of cross-validation results for standard BKT and Spectral BKT

					All opportunities		Opportunity 1		Opportunity 2		Opportunity 3+	
Model	**Par/skill**	**CV**	**AIC**	**BIC**	**Acc.***	**RMSE**	**Acc.**	**RMSE**	**Acc.**	**RMSE**	**Acc.**	**RMSE**
BKT	4	item	15380089	15478083	0.8609	0.3293	0.7469	0.4112	0.8099	0.3731	0.8740	0.3181
Spectral BKT	37	item	15853433	16758456	0.9208	0.2472	0.7472	0.4146	0.8915	0.2940	0.9337	0.2289
BKT	4	student	13947080	14045074	0.8659	0.3153	0.7435	0.4108	0.8126	0.3665	0.8799	0.3020
Spectral BKT	37	student	11553442	12458465	0.9196	0.2405	0.7469	0.4130	0.8897	0.2937	0.9325	0.2210

* For a reference, the majority class accuracy of predicting correct response for every row is 0.8569.

third digit of both accuracy and RMSE, both around 74% and 0.41 respectively. On the second opportunity prediction, Spectral BKT has a decisive edge of almost 9% and 0.08 in RMSE. On the third opportunity and further, Spectral BKT has a comfortable advantage of around 5% in accuracy and 0.09 in RMSE.

6. DISCUSSION

The performance of the Spectral BKT demonstrated a tangible improvement over standard BKT and only with an incremental change in the underlying computations. We attribute the boost in predictive performance to the several factors. First, feature compensation via considering 3-grams of original observations allows for a more stable estimate of the learning process. In a sequence of responses {0,1,0,1,1}, the third value of 0 would be treated a potential slip by the standard BKT. At the same time, Spectral BKT would consider it, as a part of the first triple {0,1,0} to be the *noisy guessing*, and then, in the second triple {1,0,1}, as part of the *noisy slipping*. Finally, in the third triple {0,1,1}, 0 would be considered to be a part of noise-free learning pattern. The fact that there are more than 2 states allows Spectral BKT to represent an intermediate configuration of student learning in addition to just known or unknown. As a result, Spectral BKT is able to deal with the noise in the observations better.

The interpretation of a new conceptualization of the process of learning remains an open question. Having agreed on that state 1 is the known state and state 4 is the unknown state, we could offer several hypotheses of what the remaining middle states are. The first hypothesis relates to the linear view of the stages of mastering the skill. When a student just started learning and only seen a few problems, their knowledge is overly specific, and they would end up guessing and failing a lot. We can call this state 3 – *too-specific*. Once the student sees more problems and starts to generalize the knowledge, they would still occasionally slip due to over-generalization. We can call this state 2 – *too-general*.

Our second hypothesis is related to a publication by Aleven and colleagues [2]. In this work, authors study the metacognitive behavior of students by administering two types of tutors. First, the cognitive tutor that implements a mastery learning approach. Second, the meta-cognitive tutor used a previously created model of effective and ineffective help-seeking behavior in order to study the effect of different meta-cognitive traits on learning. Authors conclude that the use of the standard BKT model with two states might be limiting the capability of the meta-cognitive tutor to offer effective help due to lack of intermediate states between the *known* state and *unknown* state that might give us a better insight into student behavior. In the light of the work by Aleven et al., the progression of the states could be reflecting an interaction of binary latent capturing skill mastery (*known*, *unknown*) with the binary latent capturing effective use of meta-cognitive strategies (2 mastery states * 2 metacognitive states = 4 overall states). To address this hypothesis, one might consider step durations (available in the original dataset) or design and run a focused investigation like the one in [2].

In our work, we used 3-grams of original binary observations, giving us 8 new spectral observations and we also used 4 states. This particular setup can be changed in the search of a better Spectral BKT model. Increasing the number of states could be potentially beneficial. However, one must be careful, for as the number of states grows, the chance to observe relevant patterns of binary observations drops and the Spectral BKT might be under-defined and this could have problems with performing on unseen data whether the patterns missing from the training set are present. When there are fewer states that there are spectral observations, the states serve the aggregation role. We empirically tried configurations of Spectral BKT with 2 states and 4 bigram spectral observations that did not result in an improvement over standard BKT, as well as a configuration with 8 states and 16 4-gram spectral observations that did not result in a tangible improvement over the configuration we discussed in this paper.

7. ACKNOWLEDGMENTS

This work is partially supported by the Andrew Mellon Pre-doctoral Fellowship and extends a project initiated during the Pittsburgh Science of Learning Center's Summer School at Carnegie Mellon University. We extend our special thanks to Dr. Geoffrey J. Gordon from Machine Learning Department at Carnegie Mellon University for insightful comments.

8. REFERENCES

[1] Baker, R.S.J.d., Corbett, A.T., Aleven, V. (2008) Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. EDM 2008, pp.67-76.

[2] Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16, 101-128.

[3] Gales, M. & Young, S. (2007) The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends in Signal Processing*, 1(3): 195-304.

[4] Corbett, A.T. & Anderson, J.R. (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

[5] Yudelson, M., Koedinger, K., & Gordon, G. (2013) Individualized Bayesian Knowledge Tracing Models. AIED 2013, pp. 171-180.

[6] Wang, Y. & Beck, J. (2013) Class vs. Student in a Bayesian Network Student Model. AIED 2013, pp. 151-160.

[7] Xu, Y. & Mostow, J. (2012). Comparison of methods to trace multiple subskills: Is LR-DBN best? EDM 2012, pp.41-48.

[8] González-Brenes, J.P., Huang, Y., & Brusilovsky, P. (2014). General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. EDM 2014, pp.84-91.

[9] Pardos, Z. & Heffernan, N. (2011) KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. UMAP 2011, pp. 243-254.

[10] Redner,R. & Walker,H. (1984) Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev., 26, 195-239.

[11] Beck, J. & Chang, K.-m. 2007. Identifiability: A fundamental problem of student modeling. UM 2007, 137–146.

[12] Falakmassir, M.H., Pardos, Z.A., Gordon, G.J., & Brusilovsky, P.L. (2013) A Spectral Learning Approach to Knowledge Tracing. EDM 2013, pp. 28-34.

[13] Zuk, O., Kanter, I., Domany, E., & Aizenman, M. (2006) Taylor series expansions for the entropy rate of hidden Markov processes. ICC 2006, pp.1598-1604.

Direct estimation of the minimum RSS value for training Bayesian Knowledge Tracing parameters

Francesc Martori
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona, Spain
francesc.martori@iqs.edu

Jordi Cuadros
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona Spain
jordi.cuadros@iqs.edu

Lucinio González-Sabaté
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona, Spain
lucinio.gonzalez@iqs.edu

ABSTRACT
Student modeling can help guide the behavior of a cognitive tutor system and provide insight to researchers on understanding how students learn. In this context, Bayesian Knowledge Tracing (BKT) is one of the most popular knowledge inference models due to its predictive accuracy, interpretability and ability to infer student knowledge. However, the most popular methods for training the parameters of BKT have some problems, such as identifiability, local minima, degenerate parameters and computational cost during fitting. In this paper we address some of the issues of one of these training models, BKT Brute Force. Instead of finding the parameter values that provide the lowest Residual Sum of Squares (RSS), we estimate this minimum RSS value from some a priori known values of the skill. From there we perform some preliminary analysis to improve our knowledge of the relationship between the RSS, from BKT-BF, and the four BKT parameters.

Keywords: Bayesian Knowledge Tracing · BKT Brute Force · RSS modeling

1. INTRODUCTION

1.1 Bayesian Knowledge Tracing
Bayesian Knowledge Tracing (BKT) [1] is a student model used to infer a student's knowledge given their history of responses to problems, which it can use to predict future performance. Using students' responses to questions, which are tagged with the skills that the instructor wants the students to learn, the model tells the probability a student has mastered a skill.

BKT is a two state Hidden Markov Model, these states being the one in which the student knows a given skill, and the one where the student does not. The "knowledge" state is absorbent, implying that the student will not forget the skill once it is learned. To calculate the probability that a student knows the skill given their performance history, BKT needs to know four probabilities:

L_0, the probability a student knows the skill before attempting the first problem,

T, the probability a student, who does not currently know the skill, will know it after the next practice opportunity, that is the transition probability at each practice opportunity,

G, the probability a student will answer a question correctly despite not knowing the skill,

S, the probability a student will answer a question incorrectly despite knowing the skill.

According to this model, knowledge affects performance (mediated by the guess and slip rates), and knowledge at one time step affects knowledge at the next time step: if a student is in the unknown state at time t, then the probability they will be in the "knowledge" state at time t+1 is P(T). Usually, a separate BKT model is fit for each skill and only the first attempt at each question is taken for each student.

1.2 Bayesian Knowledge Tracing – Brute Force
Bayesian Knowledge Tracing – Brute Force [2] (BKT-BF) is an algorithm to estimate the values for the BKT parameters. It is a simple brute force algorithm, where a grid of possible values is set so that for each combination of parameters, a RSS value is obtained. At the end, the combination of values resulting in the lowest Residual Sum of Squares (RSS) value for a skill is the one that will be used in BKT.
In BKT-BF, the RSS is calculated as follows:

$$RSS = \sum_{i}^{students} \sum_{t=1}^{dim} \left(O_{i,t} - C_{i,t}\right)^2 \quad \text{eq. 1}$$

Where:

$O_{i,t}$ is $\{0,1\}$ depending on the student's answer to a given question,

students is the number of different students who faced any question of a given skill,

dim is the number of different questions that are tagged with a given skill

$C_{i,j}$ is the likelihood to produce a correct answer to a question. This calculation is derived from the BKT formulas, and it is done, for the student i, as follows:

$$C_{i,t} = L_{t-1} * (1 - S) + (1 - L_{t-1}) * G \quad \text{eq. 2}$$

BKT-BF is, however, is very expensive in computational cost, as all brute force algorithms are, and does not help the identifiability [3] problem from BKT; identifiability results in different combinations of parameter values, some of which make no theoretical sense, giving similar RSS values. The other most usual algorithm is EM [4], which is not as computationally demanding but suffers from local minima issues. There are efforts to develop methods [5], [6], [7] and [8] that use different techniques to tackle the issues we mentioned, however, in this paper we will focus our work on BKT-BF.

Given that BKT-BF is an algorithm that gives good practical results, but it is so computationally expensive, the objective of this paper is to make accurate estimates of the minimum RSS value for any skill. At the same time, this might provide a better understanding of the BKT model.

2. DATA AND METHODS

The data used belongs to the 'Psychology MOOC GT - Spring 2013' dataset, accessed via DataShop (pslcdatashop.org) [9]. This course was designed by the Open Learning Initiative (OLI), who are known for their data driven design [10], [11], this fact and their long experience in course design ensure that skills have been properly tagged. The course was taken by 5615 students that issued around 2 million first attempt answers. There were 226 different skills identified in the course. The skills map used can be also found in [9]

In order to obtain the RSS values, we have used the BKT-BF algorithm. Specifically, we have used values from 0.05 to 0.95, with a 0.15 step, for L_0 and T; for G and S, the bounded approach has been taken in order to avoid model degeneracy [5], so we have used values from 0.05 to 0.30, with a 0.05 step. Given all this, 1764 different RSS values were obtained per skill.

To identify each skill, we have defined three variables:

- *dim*: number of different questions that are tagged with a given skill
- *n*: total number of responses on questions tagged with a given skill. It's the product of *students* and *dim* from eq.1
- *percent_correct (pc)*: Percentage of correct answers to questions tagged with a given skill

These variables have been chosen as they are pieces of information that one may have easy access to before computing BKT-BF.

In order to achieve the aforementioned objective, we will train a linear model using the three variables we defined for each skill. This model will allow us to make predictions of which will be any skill's minimum RSS, if we were to train it using BKT-BF. To train the model, we have extracted the minimum RSS value, resulting from the BKT-BF calculations, for each one of the skills, and used it as the RSS value for that skill. An example of the data we have worked with is shown in table 1.

It has to be noted, that skills that were tagged in less than 4 different questions (*dim*<4) have been discarded. That results in a sample of 103 different skills for training and evaluating the model.

Table 1. Data structure for skills

dim	n	pc	Skill	Grid BKT-BF	RSS min
8	4923	79,8%	1	1764 data	795.6
16	11062	89,5%	2	1764 data	1024.5
…	…	…	…	…	…

The resulting distribution of RSS values is far from being normal, as it could be expected. However, if instead of using the RSS value, we compute the Root Mean Squared Error (RMSE) for each skill, by taking the square root of the RSS divided by n, the resulting distribution is acceptably normal as we can see in the histogram shown in Figure 1 and in the Q-Q plot in Figure 2. This latter plot assesses normality by displaying the normal theoretical quantiles (x axis) and the normal data quantiles (y axis). If the distribution is perfectly normal, data would perfectly fit the dotted line.

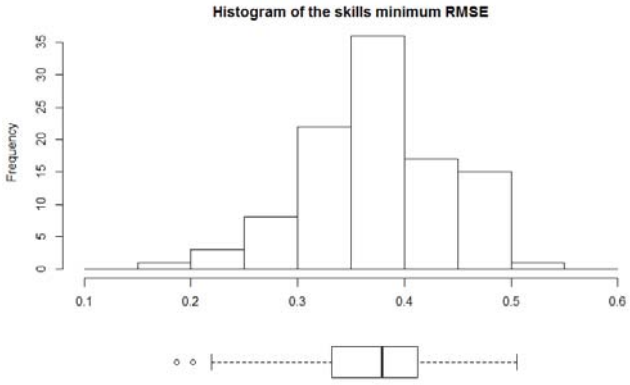


Figure 1. Histogram and boxplot of the RMSE distribution

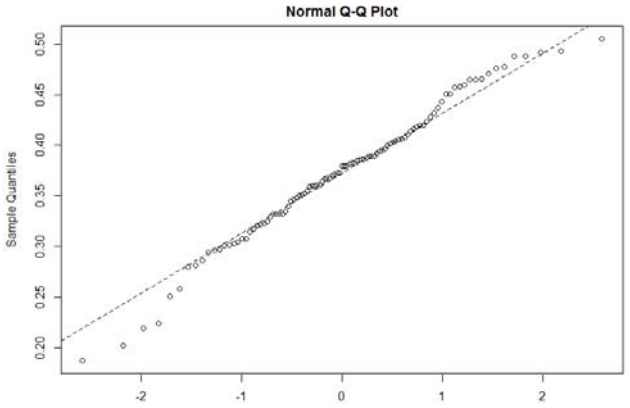


Figure 2. Q-Q plot of the RMSE

3. RESULTS

Firstly, a brief summary for the data we have worked with is shown in the table 2.

Table 2. Summary of the data for training the model.

	n	dim	pc	RMSE	RSS
min	1738	4.00	0.458	0.187	152.0
Median	5899	8.00	0.821	0.379	811.8
Mean	8556	9.65	0.807	0.373	1235.1
Max	47215	23.00	0.964	0.505	8483.4

A linear regression has been performed on the RMSE, using *n*, *dim*, *pc*, some usual transformations, such as using the logarithms and the squares of the variables, and the variable interactions as predictors. A best subset selection (using the leaps package in R, [12]) approach has been taken, resulting the best model the one using a second degree polynomial with *pc*. The results for the linear regression estimates are shown in the table 3.

Table 3. Linear regression results and error metrics

Variable	Estimate	Std. Error	t value	P(>|t|)
Intercept	0.3725	0.0009	415.9	<2e-16
pc	-0.6096	0.0091	-67.1	<2e-16
pc^2	-0.2545	0.0091	-28.0	<2e-16
Adjusted R^2		**Residual standard Error**		
0.981		0.009089		

Finally, using a random validation set (75 skills to train the model and 28 to test it), we have obtained an adjusted R^2 of 0.978, that shows a very good predictive ability for the adjusted model.

In an attempt to have a better knowledge on the relationship between the *RMSE* values and the BKT parameters, we have run a preliminary Principal Components Analysis (PCA). The resulting biplot of the PCA is shown in figure 3. For the sake of a proper understanding of the relationship between the different variables, we have eliminated the data labels from the chart. The variance explained by the first two Components of PCA is 71.4%.

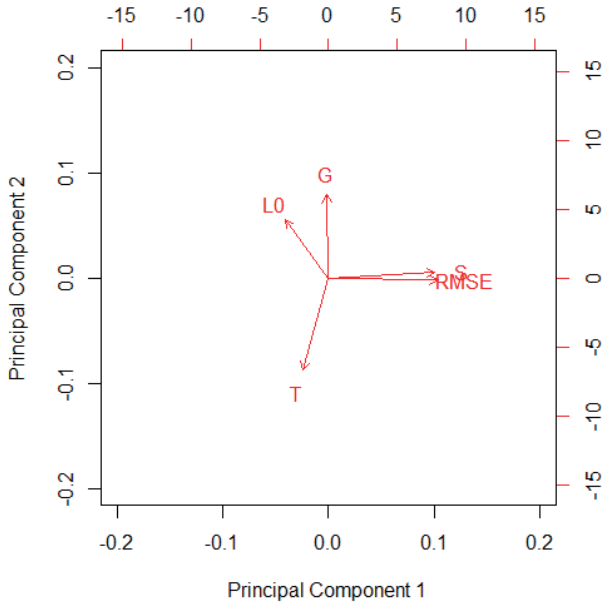


Figure 3. Plot of the PCA loadings of RMSE, L_0, T, G and S

In the chart, we can see how the RMSE is highly correlated with the slip parameter. At the same time, the parameters *G* and *T* seem to be highly inversely correlated, which is something that one can expect as the more likely it is to learn a skill, the less likely it is that you might be guessing the outcome. However, the most noticeable aspect is the orthogonality between *T*, *G* and *RMSE*. In the PCA context, orthogonality is related to poorly correlated variables. If that was to be true, it could imply that T and G have little or no effect in terms of RMSE variation. We have also calculated and drawn the biplots for each skills' RMSEs, using all BKT-BF data points, not just the minimums, and their results lead us to similar conclusions than the ones obtained from figure 3.

4. DISCUSSION AND CONCLUSIONS

We have been able to find a linear model that allows us to estimate the minimum RSS value for the training of the BKT parameters. Using this, we might be able to find a quicker convergence using a modified version of BKT-BF, so that the computational cost will be reduced. Even though that the model has been developed using the RMSE instead of the RSS, the model will also be useful for predicting the latter as the only difference is a transformation involving *dim* and *n*.

We are aware that, in the BKT-BF calculations, we are using a step much larger than the one recommended by the algorithm. This shouldn't be a problem with the conclusions we reached because we are not using BKT-BF for estimating the BKT parameters, but to generate data with which we train a model for estimating the minimum RSS for any skill.

The very high performance of the model, in terms of adjusted R^2, may be indicating that BKT works better when the percentage of correct answers is very high, as the RSS decreases. This has some implications in the BKT model because if the percentage of correct answers is very high, there might not be much room for T and G in the model. We would only be trying to adjust the probability of already knowing the skills before doing the course and the probability of slipping.

To be more certain about the conclusions stated here, the following steps have to include using, at least, a different dataset to shed some light around the suspicions that arise on the influence of T and G in the BKT model. A deeper analysis beyond an exploratory PCA is also required.

5. AKNOWLEDGEMENTS

We acknowledge the help received from colleagues at OLI at Stanford University.

6. REFERENCES

[1] Corbett, A. T. and Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253-278.

[2] Baker, Corbett, Gowda, Wagner, MacLaren, Kauffman, Mitchell, & Giguere, 2010. Bayesian Knowledge Tracing Brute Force model fitting code. http://users.wpi.edu/~rsbaker/edmtools.html

[3] Beck, J. E., Chang, K. M. 2007 Identifiability: A fundamental problem of student modeling. In: Conati, C., McCoy, K., Paliouras, G. (Eds.) UM 2007. LNCS, vol. 4511/2007, pp. 137- 146.

[4] Moon, T. K. 1996. The expectation–maximization algorithm. IEEE Signal Process. Mag., 13, 47–60

[5] Baker, R.S.J.d., Corbett, A. T., Aleven, V.: More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Woolf, B., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) ITS 2008. LNCS, vol. 5091/2008, pp. 406-415. Springer, Berlin Heidelberg (2008)

[6] Hawkins, W., Heffernan, N.T., Baker, R.S.J.d. (2014) Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. Lecture Notes in Computer Science Volume 8474, 2014, pp 150-155.

[7] Pardos, Z. A., Heffernan, N. T.: Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. In: Baker, R.S.J.d., Merceron, A., Pavlik, P.I. (Eds.) Proceedings of the 3rd International Conference on Educational Data Mining, pp. 161-170 (2010)

[8] Rai, D., Gong, Y., Beck, J.: Using Dirichlet priors to improve model parameter plausibility. In: Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.) Proceedings of the 2nd International Conference on Educational Data Mining, pp. 141-150 (2009)

[9] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2010. A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

[10] Strader, R., Thille, C.: The Open Learning Initiative: Enacting Instruction Online. In Oblinger D. G. (Ed.), Game Changers. Education and Information Technologies, pp. 201-213 (2012)

[11] Thille, C. (2012). Changing the Production Function in Higher Education. Making Productivity Real. American Council on Education 2012.

[12] Thomas Lumley using Fortran code by Alan Miller (2009). leaps: regression subset selection. R package version 2.9. http://CRAN.R-project.org/package=leaps

Goodness of fit of skills assessment approaches: Insights from patterns of real vs. synthetic data sets

Behzad Beheshti
Polytechnique Montreal
behzad.beheshti@polymtl.ca

Michel C. Desmarais
Polytechnique Montreal
michel.desmarais@polymtl.ca

ABSTRACT

This study investigates the issue of the goodness of fit of different skills assessment models using both synthetic and real data. Synthetic data is generated from the different skills assessment models. The results show wide differences of performances between the skills assessment models over synthetic data sets. The set of relative performances for the different models create a kind of "signature" for each specific data. We conjecture that if this signature is unique, it is a good indicator that the corresponding model is a good fit to the data.

1. INTRODUCTION

There exists a large array of models to represent and assess student skills. Item Response Theory (IRT) is probably the most established method. It dates back to the 1960's and is still one of the prevailing approaches (see [1]). But many other methods have been introduced in recent years. Among them is the family of models that rely on slip and guess factors [12, 11], such as the DINA (Deterministic Input Noisy And-Gate), DINO (Deterministic Input Noisy Or-Gate), and other variants (see [7]). Other approaches are based on the Knowledge Space theory of Doignon and Falmagne [10, 8], which does not directly attempt to model underlying skills but instead rely on observable items only. Finally, recent methods based on matrix factorization have also emerged in the last decade [16, 15, 5, 2]. They factorize the student per item results matrix into the linear product of the so called Q-matrix (skills required per item) and the skills mastery matrix.

We undertook the effort of comparing prevailing and widely different methods to assess skills. The comparison is based on each method's ability to predict item/task outcome. However, in addition to providing a comprehensive comparison of skills assessment approaches, this research also aims to develop a method that uses synthetic data to characterize item outcome data and yield insights about this data's ground truth structure. Beyond the obvious expectation that the model behind the generation of synthetic data will outperform all others on this data set, we conjecture that the relative performance of all other methods will be unique and can represent a kind of "performance signature" that characterizes this type of data. Therefore, if a data set from a real setting reflects that signature, it would constitute a good indicator that the corresponding model is a good fit.

This work is an extension of [3], and is similar in its general principles to the approach of Rosenberg-Kima and Pardos [13], who take the likelihood of a model's parameter space as a signature instead of the performance of different techniques as we do here. Their idea is that the likelihood function of two parameters of Bayesian Knowledge tracing is a unique characterization of a data set. If the likelihood function of synthetic data generated with estimates of these parameters from real data has the same "signature" as the likelihood function of that real data, then the model is a good fit.

2. SKILLS ASSESSMENT METHODS

We compare a total of seven different skills assessment methods. We briefly describe them here and refer the reader to [7] and [6] for details. They can be grouped into four categories:

(1) The single skill Item Response Theory (IRT) approach. IRT is a well known framework based on logistic regression and represents student proficiency by a single skill (although we also find multiple skills version of IRT, MIRT).

(2) The POKS (Partial Order Knowledge Structrures) represents the order in which items are learned and uses a Naive Bayes framework to make inferences based on this order. It does not represent latent skills, but a Q-matrix can be used aposteriori on the estimated item outcome to assess skills.

(3) The matrix factorization approach decomposes the matrix of m students by n items into the product of m students by k skills representing the latent skills assessment, and an k by n Q-matrix.

(4) The multi-skills family of DINA/DINO approaches are equivalent to a binary matrix factorization framework, where the skill outcome is a boolean product of binary vectors, but they also contain *guess* and *slip* parameters. In the DINA version, the boolean product is based on the AND operator, whereas DINO is based on the OR operator.

Finally, as a baseline for comparison we also consider the *Expected value* as the simplest model. It takes into account the mean item difficulty and student ability to compute the expected score of the corresponding item. The mean difficulty is the average success rate of an item obtained from the training data, while the student ability is the mean success rate obtained from the observed data. The Expected value is the geometric mean of the product of these two means.

3. METHODOLOGY

The performance of each method is assessed on the basis of 10-folds cross-validation, and on observing all items from a student except the one that is to be predicted. For each fold, each item in the set is taken as a target prediction once.

For the IRT and POKS models, the parameters of each models are trained and the testing is based on feeding the models with all but one question. A probability of mastery is obtained and rounded, resulting in a 0/1 error loss function. We report the mean accuracy as the performance measure. The R package `ltm` is used for parameter and skills estimation.

For the other models, they rely on a Q-matrix to estimate the remaining item outcome. For the linear conjunctive and compensatory models, the Q-matrix needs to be normalized such that if all skills for an item are mastered, the inner product of the skills mastered vector and the skills required will be 1. Here too, results are rounded for obtaining a 0/1 loss function. Normalization of the Q-matrix is not necessary for the DINA and DINO models.

4. DATA SETS AND SYNTHETIC DATA GENERATION

The performance of the methods is assessed over a total of 14 data sets, 7 of which are synthetic, and 7 are real data. They are listed in table 1), along with the number of skills of their Q-matrix, their number of items, the number of the student respondents, and the average score. Table 1 also reports the Q-matrix used. To make these data sets more comparable to their real counter part we used Q-matrices and other parameters from real data sets to generate synthetic datasets.

Of the 7 real data sets, only three are independent. The other 4 are variations of a well known data set in fraction Algebra from Tatsuoka's work [14]. The real data sets were obtained from different sources and are freely available from the CDM and NPCD R packages. The Q-matrices of the real data sets were made by experts.

The synthetic data sets are generated from their underlying respective skills assessment model.

For POKS, the structure was obtained from the Fraction data set and the conditional probabilities were generated stochastically, but in accordance with the semantic constraints of these structures and to obtain an average success rate of 0.5.

For IRT, the student ability distributions was obtained from the Fraction data set, and the item difficulty was set to reasonable values: averaging to 1 and following a Poisson distribution that kept most values between 0.5 and 2[1].

Data set	Number of			Mean Score	Q-matrix
	Skills	Items	Students		
Synthetic					
1. Random	7	30	700	0.75	$\mathbf{Q}_{01}$
2. POKS	7	20	500	0.50	$\mathbf{Q}_{02}$
3. IRT-Rasch	5	20	600	0.44	$\mathbf{Q}_{04}$
4. DINA	7	28	500	0.31	$\mathbf{Q}_{5}$
5. DINO	7	28	500	0.69	$\mathbf{Q}_{6}$
6. Linear Conj.	8	20	500	0.24	$\mathbf{Q}_{1}$
7. Linear Comp.	8	20	500	0.57	$\mathbf{Q}_{1}$
Real					
8. Fraction	8	20	536	0.53	$\mathbf{Q}_{1}$
9. Vomlel	6	20	149	0.61	$\mathbf{Q}_{4}$
10. ECPE	3	28	2922	0.71	$\mathbf{Q}_{3}$
Fraction subsets and variants of $\mathbf{Q}_1$					
11. 1	5	15	536	0.53	$\mathbf{Q}_{10}$
12. 2/1	3	11	536	0.51	$\mathbf{Q}_{11}$
13. 2/2	5	11	536	0.51	$\mathbf{Q}_{12}$
14. 2/3	3	11	536	0.51	$\mathbf{Q}_{13}$

Table 1: Datasets

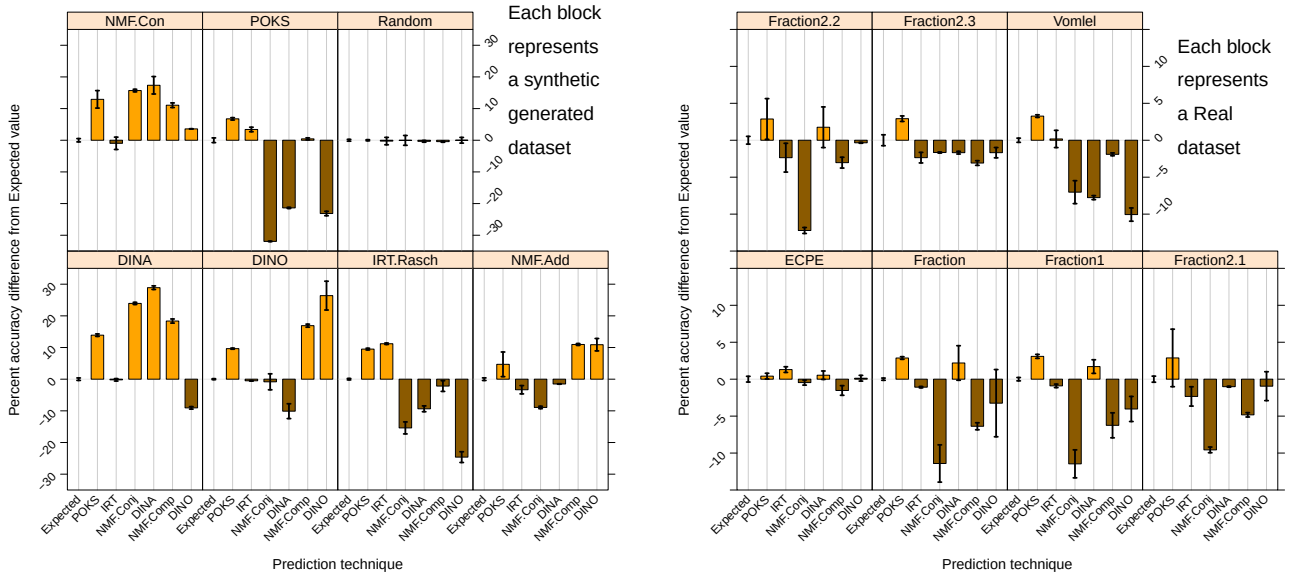
The matrix factorization synthetic data sets of DINO and DINA were generated by taking a Q-matrix of 7 skills that contains all possible combinations of 1 and 2 skills, which gives a total of 28 combinations and therefore the same number of items. Random binary skills matrix were generated and the same process was used for both the DINO and DINA data sets. Item outcome is then generated with a slip and guess factor of 0.1.

A similar process was followed to generate the Q-matrices and the skills matrices $\mathbf{S}$ of the linear matrix factorization data sets

Note that the first 3 models do not rely on any Q-matrix for the data generation process, but the DINO/DINA and matrix factorization assessment methods still require one. To define these Q-matrices (denoted $\mathbf{Q}_{0x}$ in table 1, a wrapper method was used to first determine the number of skills according to [4], then a Q-matrix was derived with the ALS method (see [9]).

All data sets are considered *static* in the sense that they represent a snapshot of student test performance data. This corresponds to the assumption that the student has not mastered new skills during the process of assessment, as we would expect from data from learning environments. This assumption is common to all models considered for this study.

[1]Done by generating random numbers from a Poisson distribution with lambda parameter set to 10 and dividing by 10.



(a) Synthetic datasets

(b) Real dataset

Figure 1: Item outcome prediction accuracy results. Each plot reports the prediction accuracy of the different techniques, whereas each bar shows the percentage difference in accuracy from the Expected value baseline (square root of item × student average success rates).

5. RESULTS AND DISCUSSION

Figure 1 shows the difference between the performance of each technique and the Expected value accuracy as computed by the geometric mean: square root of item × student average success rates. An error bar of 1 standard deviation is reported and computed over the 10 random sampling simulation runs and provides an idea of the variability of the results. Also reported is the performance of random data with a 0.75 average success rate.

As expected, when the generative model behind the synthetic data set is the same as the skills assessment technique, the corresponding technique's performance is generally the best. Exceptions are found for the linear conjunctive case, where the corresponding technique performance comes second. For real data, the performance of many techniques is often lower than the Expected value baseline. This is likely due to the fact that all but one item is observed, the target, and therefore the Expected value is a reliable predictor.

The most consistent performance across the synthetic data sets are those of POKS and IRT, with POKS showing a greater accuracy on average. This consistency also transfers to the real data sets, although the differences are smaller and the Expected value method performance is sometimes better than the IRT one. But as mentioned the good performance of the Expected value may well depend on the relatively high number of observations for each data sets (1 less than the total number of questions per data set).

Also worth noticing is that the random data set has a flat performance across techniques which corresponds to the dominant class prediction. This is not necessarily surprising, but it is reassuring in a sense to know that they all perform the same in the face of random data and this performance is indeed the best that could be obtained.

For the independent real data sets, the differences between techniques are less divergent and closer to the Expected value technique, although the best performers are still significantly better than the Expected value for the Fraction (POKS and DINA) and Vomlel (POKS) data sets. However, for the ECPE data set, the pattern corresponds closely to that of random data: The Expected value performance is close to the dominant class performance, and all techniques are aligned towards this performance. One possibility is that all student perform more or less the same and therefore no technique is good at discriminating high/low performers.

The results from the subsets of the Fraction data shows that the pattern of the Fraction performance data set repeats over Fraction-1, Fraction-2/1 and Fraction-2/2, in spite of the different number of skills and different subsets of questions. However, it differs substantially from Fraction-2/3 for the NMF conjunctive performance which reaches that of the NMF compensatory one. This is readily explained by the fact that the Q-matrix of this data set has the property of assigning a single skill to each item, in which case the two matrix factorization techniques become equivalent.

As mentioned, the performance of the Expected value technique is high for real data, and systematically close to the best performers, POKS and DINA, which only have 2–4% better performance than the Expected value. Note that this is still substantial because we have to look at this difference relative to the remaining error (about 20%), but it is far less than for the synthetic data sets, especially on a relative difference basis.

6. CONCLUSION

This study relies on the assumption that better skills models result in better item outcome prediction. The results do show wide differences in the performance of the techniques for different synthetic data sets. For real data sets, the differences are smaller, though still significant, especially in terms of relative residual errors. Based on the results, we could conclude that POKS and DINA would provide more accurate estimates of skills.

Let us return to the comparison of real vs. synthetic data and to the conjecture that this comparison can help determine whether a specific skill model corresponds to the ground truth of some data set. This is a complex question but some clear hints are given in the results. There is a clear evidence in the DINA vs. DINO performance of figure 1 data that, if a Q-matrix is conjunctive vs. disjunctive, the results show a much better the fit to the corresponding model. Evidence is also some evidence to the claim that unidimensional data sets, i.e. a domain for which a single skill best characterizes the performance data, are best modelled by the IRT single skill IRT or the skill-less POKS models, and the multi-skills NMF conjunctive and DINA approaches do rather poorly. Conversely, multiple skills data sets of the DINO/DINA and linear family of models are better characterized by multi-skills approaches, and the IRT single skill performance is much lower in relative terms.

Another interesting finding is that random data does have a signature of its own: all methods converge towards the score of the majority class. Now, this result could stem from a set of highly similar response patterns from students, but it is clearly different from, for example, the Fraction-2/3 data set, for which all methods have relatively similar performance but they are all well above the majority class condition (AVG Success rate).

Therefore, we do conclude that there is evidence to support the claim that the relative performance of the different skills modelling approaches do create signatures over data sets and can yield some evidence about the ground truth. And if we accept this perspective, then we can also conclude that the real data sets we studied do not correspond to any of the prototypical synthetic data sets. The ground truth may involve correlations between skills, which we did not take into account. Or, the Q-matrices we have studied are not faithful to the reality and, for example, may involve combinations of conjunctive and disjunctive skills. In fact, many explanations can be evoked, but the hope is that by looking at the relative performances of each method we can gain some insights of the best explanations.

References

[1] F. B. Baker and S.-H. Kim. *Item Response Theory, Parameter Estimation Techniques (2nd ed.)*. Marcel Dekker Inc., New York, NY, 2004.

[2] T. Barnes. Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on Educational Data Mining*, 2010.

[3] B. Beheshti and M. C. Desmarais. Predictive performance of prevailing approaches to skills assessment techniques: Insights from real vs. synthetic data sets. In *5th International conference on Educational Data Mining, EDM 2014*, pages 409–410, London, UK 2014.

[4] B. Beheshti, M. C. Desmarais, and R. Naceur. Methods to find the number of latent skills. In *5th International conference on Educational Data Mining, EDM 2012, Chania, Greece, 19–21 June 2012*, pages 81–86. Springer, 2012.

[5] M. Desmarais. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *4th International Conference on Educational Data Mining, EDM*, pages 41–50, 2011.

[6] M. C. Desmarais, B. Beheshti, and R. Naceur. Item to skills mapping: Deriving a conjunctive Q-matrix from data. In *11th Conference on Intelligent Tutoring Systems, ITS 2012*, pages 454–463, Chania, Greece, 14–18 June 2012 2012.

[7] M. C. Desmarais and R. S. d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[8] M. C. Desmarais, P. Meshkinfam, and M. Gagnon. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, 16(5):403–434, 2006.

[9] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices. In *6th International Conference, AIED 2013, Memphis, TN, USA*, pages 441–450, 2013.

[10] J.-P. Doignon and J.-C. Falmagne. *Knowledge Spaces*. Springer-Verlag, Berlin, 1999.

[11] B. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric IRT, Dec. 27 2000.

[12] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[13] R. B. Rosenberg-Kima and Z. Pardos. Is this data for real? In *Twenty Years of Knowledge Tracing Workshop (colocated with EDM 2014*, pages 141–145, London, UK.

[14] K. K. Tatsuoka. *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.

[15] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In A. Verbraeck, M. Helfert, J. Cordeiro, and B. Shishkov, editors, *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011*, pages 69–78. SciTePress, 2011.

[16] T. Winters, C. Shelton, T. Payne, and G. Mei. Topic extraction from item level grades. In *American Association for Artificial Intelligence 2005 Workshop on Educational Datamining*, 2005.

A Transfer Learning approach for applying Matrix Factorization to small ITS datasets

Lydia Voß
Information Systems and Machine Learning Lab
Universitätsplatz 1, 31141
Hildesheim, Germany
lvoss@ismll.uni-hildesheim.de

Carlotta Schatten
Information Systems and Machine Learning Lab
Universitätsplatz 1, 31141
Hildesheim, Germany
schatten@ismll.uni-hildesheim.de

Claudia Mazziotti
Institute of Educational Research, Ruhr-University Bochum
Universitätsstraße 150, 44780
Bochum, Germany
claudia.mazziotti@rub.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
Universitätsplatz 1, 31141
Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

Machine Learning methods for Performance Prediction in Intelligent Tutoring Systems (ITS) have proven their efficacy; specific methods, e.g. Matrix Factorization (MF), however suffer from the lack of available information about new tasks or new students. In this paper we show how this problem could be solved by applying Transfer Learning (TL), i.e. combining similar but not equal datasets to train Machine Learning models. In our case we obtain promising results by combining data collected of German fractions' tasks (517 interactions, 88 students, 20 tasks) with their non-exact translation of a previously American US version (140 interactions, 14 students, 16 tasks). In order to do so we also analyze the performance of MF based predictors on smaller ITS' samples evaluating their usefulness.

Keywords

Transfer Learning, Intelligent Tutoring Systems, Matrix Factorization, Vygotsky Policy Sequencer

1. INTRODUCTION

One of the main uses of Educational Data Mining in Intelligent Tutoring Systems (ITS) is Performance Prediction, which aims to ameliorate the student's model by understanding whether a student mastered a specific set of skills or not. Specific methods, e.g. Matrix Factorization (MF), suffer from the lack of available information about new ITS tasks or new students imposing challenging requirements on organizing trials. This happens because the algorithm is personalized, i.e. there is one model for each student interacting with the system and one for each task one can practice with. If no data are available for one task or for one student no prediction can be computed, this problem is called the cold-start problem. Moreover, first data for new tasks in ITS applications are obligatorily collected in a specific sequence, which is generally fixed or rule-based. As a consequence more interaction data are available for the first tasks in the sequence whereas just a few are available for the last ones making the prediction for specific tasks more challenging. In the FP7 iTalk2Learn project[1] we developed a domain independent sequencer [9] for one of our use cases based on MF Performance Prediction. One of this use cases is a German translation of Fraction Tutor (FT) a web-based Cognitive Tutor for fractions developed by Carnegie Mellon University[2]. Our data collection for the German version (88 students, 20 tasks, 517 interactions) represents, to the best of our knowledge, one of the smallest dataset used to train a MF based recommender for Performance Prediction in ITS. We also possess the data collected with the original US American version (16 tasks, 14 students and 140 interactions), which, according to common practice, should be discarded. In this paper we want to:

- Show, that we can use two different but comparable datasets (the German and English ones) to ameliorate Performance Prediction.
- Analyze in detail the effects of a small dataset on the performances of MF used as performance predictor.
- Propose a practical solution to the data collection to reduce data sparsity.

The paper is structured as follows. the second and third section describe the state of the art and the theory behind the performance predictors we used. In Sec. 4 the data collection, translation and preprocessing is described. In the Experiment Section we discuss the usefulness and measure

[1] www.iTalk2Learn.eu
[2] https://mathtutor.web.cmu.edu/

the performances of MF based predictors. Then we conclude the Section combining the English and German datasets to evaluate the feasibility of Transfer Learning approaches to exploit generally discarded data in ITS.

2. RELATED WORK

As we did not have access to the required skills information in [7, 8], MF and the VPS sequencer presented in [9] are used for Performance Prediction. MF has many applications, its most common use is for Recommender Systems and recently this concept was extended to Performance Prediction and to sequencing problems in ITS [10, 9], but all experiments were done with simulated students' interactions or offline experiments. In [7], we showed how the VPS sequencer could be integrated and worked in a large commercial ITS. A similar analysis on MF was done in [5] where Performance Prediction was tested on a small dense dataset (each student saw each task). The performance predictors were standard Collaborative Filtering techniques, where the best one performing resulted to be Biased Matrix Factorization (see Section 3.1 for more details). In this paper, we possess even less interactions. Not only the students did not interact with all available tasks, but sometimes they also solved less than three tasks. We try to solve this problem with Transfer Learning (TL)[3]. In contrast to classical Machine Learning methods, TL methods exploit the knowledge accumulated from auxiliary data to facilitate predictive modeling consisting of different but similar patterns in the current data [2]. Auxiliary data could mean additional information describing the state of the system and/or data collected with a second slightly modified version of the same system (e.g. using equal movies from different movie rating datasets and transfer the knowledge [4]). In this case correctly done transfer of knowledge, i.e. using similar but not equal datasets, is required and could improve the performance of predictors in classification and regression tasks ([4]) by considering previously unused data. This approach becomes particularly helpful when recollection is expensive or impossible. However TL was never applied to ITS data. Consequently, in Sec. 5.3 we evaluate the feasibility of applying TL to our use case to get a better Performance Prediction.

3. MATRIX FACTORIZATION BASED PREDICTORS

We use MF to predict the students performance. The matrix $Y \in \mathbb{R}^{S \times T}$ can be seen as an incomplete table of T tasks and S students. This matrix is used to train the system. MF is the approximation of this incomplete matrix by decomposing it in two smaller matrices $W \in \mathbb{R}^{S \times K}$ and $H \in \mathbb{R}^{T \times K}$. The elements of the two matrices are called *latent features* and are learned with gradient descend.

Using the available entries (e.g. the score recorded from previous tasks) the missing entries can be computed by means of very fast optimization algorithms. In our experiments we use MF and a simple variation of MF, the Biased Matrix Factorization (BMF) which uses three additional variables: the global average performance μ, the student (user) bias b_s and the task (item) bias b_t. For predicting students performance the following equation is used (for MF without the bold variables):

$$p_{t,s} = \boldsymbol{\mu} + \boldsymbol{b_s} + \boldsymbol{b_t} + \sum_{k=1}^{K} w_{sk} h_{tk}, \tag{1}$$

t represents a task, s a student, k the latent features and K represents the total number of latent features. The optimization function is represented by:

$$\min_{w_s, h_t, \boldsymbol{b_t}, \boldsymbol{b_s}} \sum_{s,t \in \mathcal{D}} (y_{ts} - \hat{y}_{ts})^2 + \lambda \left(\|W\|^2 + \|H\|^2 + \|\boldsymbol{b_t}\|^2 + \|\boldsymbol{b_s}\|^2 \right) \tag{2}$$

with $\mathcal{D}$ the set of collected task student interactions. The final goal of the algorithm is to minimize the Root Mean Squared Error (RMSE) on the set of known scores.

In order to evaluate the performances of BMF and MF generally simple models like Global Average (GA, using the Global Average Score (GAS) of the students as prediction value) are used. To check which is the contribution of the Biases of the BMF to the performance of the MF we use the model called Biases, which has Eq. 2 as optimization function and Eq. 1 as prediction function, but with $K = 0$.

4. DATA COLLECTION AND ITS CHARACTERISTICS

In this section we describe the ITS we used, the data collection and what was done to connect Fraction Tutor and MF approaches.

4.1 Data collection and sequencing

We have carefully translated the English/US American FT tasks into child-friendly German and iteratively adapted to German students' needs. As a result of the translation and adaption process the US American and the German tasks are not 100% identical and we are using TL according to the definition in Sec. 2 and exploiting the knowledge from the auxiliary Englis dataset to ameliorate the German Performance Prediction.

We used three different sequences to have an equal number of interactions for each task, each sequence using a different order of task categories (6 categories). The interleaved sequence starts with one task of each category (hierarchically) and repeats this process. The second sequence refers to the so called blocked practice sequence where first all tasks of category I need to be solved, then category II and so on. Last is the mixed sequence that has a coincidental order.

In order to collect log data and train the MF for the FT we conducted a study with students (i.e. fifth graders) in classrooms (i.e. 21-28 students per class) in Germany. Students of three classes (88 students) of a German Gymnasium could interact with FT which was integrated in the iTalk2Learn platform [4].

The US American data were collected when students (14 of one class) interacted with the US American version of FT [3]. To these students tasks were proposed in a single sequence. All of them completed at least half of the sequence.

4.2 Dataset characteristics

[3]From now on we will refer to Machine Learning's Transfer Learning as TL in order not to mix it with the students' transfer learning

[4]The iTalk2Learn platform is a Plug-In platform used to integrate different components. In our case: FT tasks, database, and simple fixed sequencer.

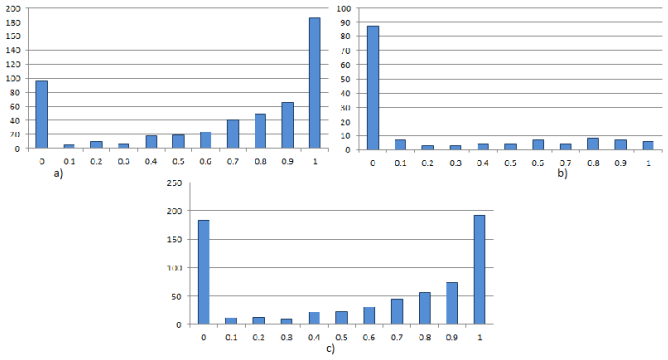


Figure 1: a) German scores b) English scores, c) combined German and English scores

For exploring the task cold-start problem for the German and English datasets (described in Sec. 1) we assigned to each task IDs from 0 to 23, where German and English tasks' (0-15) translations have the same ID. As a result we have: 14 interactions for IDs $0-6$, 11 for ID 7 $((7;11))$, $(8;10)$, $(9;8)$, $(10;6)$, $(11;2)$, $(12;2)$, $(13;1)$, $(14;1)$, $(15;1)$. For the German data the interactions are more spread out because of the three different sequences which were used: $(0;38)$,$(1;59)$, $(2;36)$, (3;0), (4;73), (5;47), (6;5), (7;0), (8;22), (9;29), (10;3), (11;0), (12;22), (13;32), (14;0),(15;0), (16;24), (17;32), (18;12), (19;26), (20;29), (21;28), (22;0), $(23,2)$. There are IDs only used in the English data: $(3,7,11,15)$. The tasks (11, 14, 15, 22, 23) have less than 2 interactions for the German and English datasets and are removed in the preprocessing. Thanks to the different sequences we have a sufficient number ([6]) of interactions for most tasks. For the English experiments we removed the last tasks, since there were too few intercations.

For the students' cold-start problem the dataset can be considered as sparse. The English dataset should be less influenced by the students' cold-start problem, because each student interacted at least with 7 tasks.

In order to have a continuous score measure as we had in [9] we used following equation to compute the score:

$$\text{score} = 1 - \left(\frac{\#hints}{\#totalnumhints} + (\#incorrect * 0.1) \right) \quad (3)$$

If the score is less than zero we set the score to 0 avoiding negative scores. For the German (a)), English (b)) and German+English (c)) data we computed the score Histogram to measure how much the data is unbalanced (See Fig. 1). Both datasets are very unbalanced but by combining the two datasets we can achieve a more balanced distribution. We will explain in the Experiment Section how this is influencing the models' performances.

5. EXPERIMENTS

To split the data in test and train set we used Leave One Out (LOO) for each student; which is a common approach to split for small datasets (here we used the last task seen by the student). To evaluate the error we measure the RMSE averaged over five experiments to avoid the influence of the random initialization of the model parameters on the model performances. The standard deviation of the error for the models prediction lies around 10^{-3}, which is normal for

HL	GA	Biases	MF	BMF
≥ 3	0.337	0.390	0.405	0.386
≥ 4	0.336	0.371	0.385	0.370
≥ 5	0.325	0.325	0.337	0.334
≥ 6	0.319	0.321	0.328	0.322
≥ 7	0.333	0.358	0.355	0.355
≥ 8	0.345	0.298	0.296	0.292

a)

HL	GA	Biases	MF	BMF
≥ 7	0.285	0.240	0.235	0.235
≥ 8	0.295	0.241	0.229	0.218

b)

Figure 2: a) RMSE German, b) RMSE English

HL	≥ 3	≥ 4	≥ 5	≥ 6	≥ 7	≥ 8
GAS	0.673	0.673	0.673	0.673	0.673	0.673
# test students	88	72	53	38	26	22
# interactions	517	501	482	467	455	451

Table 1: GA and test size German data

movie recommender datasets and small datasets. For each experiment we used the models described in sec. 3 (GA, MF, BMF). For finding the best hyperparameters we used Grid Search (learning rate: $[0.01, 0.09]$ stepsize 0.01; regularization: $[0.001, 0.009]$ stepsize 0.001, $[0.01, 0.09]$ stepsize 0.01, $[0.1, 0.9]$ stepsize 0.1; num. iterations: $100-300$ stepsize 20; num. latent features: $2-100$ stepsize 10). Moreover for each experiment we computed the performance Global Average Score (GAS) and report the number of students whose data are used.

5.1 Cold-start problem, MF Utility and Intra-Student Variance

For our experiments we studied different History Lengths (HL), i.e. the number of interactions the student had with the ITS, and we deleted the students with a HL less than 2. Starting with HL ≥ 3 we continued removing the students with $HL \leq 4$, $HL \leq 5$, etc. until $HL \leq 8$. We kept the same train data and just removed the test data, so the test set shrinks while increasing the HL requirements. GAS and number of test students are reported in Tab. 1. Table a) in Fig. 2 lists the RMSE for the German dataset.

The performances as well as the behavior of Biases, BMF and MF are coherent with the one reported in [10]. For $HL \leq 5$ Biases, MF and BMF have not sufficient informationto predict the performances (see a) in fig. 2). Keeping students with $HL \leq 5$ in the train influenced BMF negatively. The small gain between BMF and Biases can be explained with the performances of MF which are almost always worse than GA ones. This is coherent with MF and BMF behaviors where generally Biases give a strong contribution to the model performances. We can say that the Performance Prediction of GA was positively influenced by having all data in the train set, since it can be computed on a more robust statistic. BMF and MF are in general influenced by data of students with short history negatively at the beginning, although, for students with a longer history, these data can be used to ameliorate performances.

Next we evaluate the performances of Biases/MF/BMF on an even smaller dataset: the English one. The performances also of GA are quite good, although Biases, MF, and BMF clearly outperform it (see b) in Fig. 2). GA prediction ability is due to the fact that the dataset is highly unbalanced; with a majority of samples with 0 score the probability that a sample of this dataset is similar to the GAS is higher. Fig. 2 shows that BMF outperforms the Biases and the re-

HL	GA	Biases	MF	BMF
≥ 3	0.506	0.391	0.407	0.389
≥ 4	0.500	0.375	0.389	0.375
≥ 5	0.522	0.333	0.342	0.331
≥ 6	0.516	0.322	0.336	0.321
≥ 7	0.523	0.346	0.362	0.344
≥ 8	0.514	0.293	0.285	0.288

a)

HL	GA	Biases	MF	BMF
≥ 7	0.564	0.277	0.288	0.273
≥ 8	0.564	0.283	0.310	0.275

b)

Figure 3: a) RMSE GerEng, b) RMSE EngGer

sults are better than the German ones. According to our previous experience, we think that the difference in the performances (comparing experiments with same HL to avoid the cold-start problem contribution) is due to the variance between the different elements of the students' population under study. In our previous work [1] we showed the negative impact of intra-class variance in the performance of classifiers with small data samples. This applies in our opinion to the case because the intra-student variance of the German data, collected in three classes from different schools, should be higher than the intra-student variance of the English dataset that was collected in one class only.

5.2 Transfer Learning

To test the possibility to use English data to ameliorate the German prediction performances, we combined the English and German datasets as follows. In this experiment the data from an English task and its translation are considered by the MF as the same task. When combining the German and English datasets (See Table a) in Fig. 3), the performances of GA drop to approximately 0.5 because the most samples are almost equally distributed between 0 and 1 with a GAS around 0.56. To prove feasibility of TL we ran more experiments starting with the best results of the previous Sections. We added the English data to the German train set Table a) in Fig. 3), where the addition of the English data in training is always taking to a contribution for $HL \geq 6$.
The same amelioration cannot be seen when adding the German data to the English train, since adding the German data increases the intra-student variance worsening the English model performances (Table b) in Fig. 3, and Tab. 2).

BMF + HL	≥ 3	≥ 4	≥ 5	≥ 6	≥ 7	≥ 8
German	0.386	0.370	0.334	0.322	0.355	0.292
GerEng	0.389	0.375	0.331	0.321	0.344	0.288
English	/	/	/	/	0.235	0.218
EngGer	/	/	/	/	0.273	0.275

Table 2: Comparison of BMFs perfromances for all experiments.

6. CONCLUSIONS

In this paper we proposed a practical solution to the data collection to reduce data sparsity, by proposing tasks with different sequences. Moreover, we analyzed in detail the effects of a small dataset on the performances of MF used as performance predictor. Thanks to these analyses it was also possible to determine the utility of MF based performance predictors and sequencing in new ITS' tasks. Considering the Utility of BMF in comparison to GA, before having at least 7 interactions for a student it would be better to use GA as performance predictor. With using TL we already get better results for BMF with HL ≥ 5. This should hold theoretically also for the use of the VPS, although an experiment with online model update is required for a full evaluation. Finally, we proposed to exploit generally discarded data exploiting the concept of TL. As future work we will investigate more advanced methods to perform TL on small datasets and try to ameliorate performances of the first BMF predictions ($HL \leq 5$).

7. ACKNOWLEDGMENTS

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 318051 - iTalk2Learn project (www.iTalk2Learn.eu). Special thanks goes to Jenny Olsen and Martina Rau who agreed to give us access to their versions of Fractions Tutor. Special thanks further goes to Brett Leber and Jonathan Sewall who helped us by adapting the Fractions Tutor tasks into German. Last but not least, we want to thank Hamza Sati, Sebastian Strauss, Jörg Striewski, and Richard Hesse for supporting us when conducting the classroom study in Germany.

8. REFERENCES

[1] R. Janning, C. Schatten, and L. Schmidt-Thieme. Hnnp-a hybrid neural network plait for improving image classification with additional side information. In *ICTAI*. IEEE, 2013.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang. Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 2015.

[3] J. Olsen, D. Belenky, V. Aleven, and N. Rummel. Using an intelligent tutoring system to support collaborative as well as individual learning. In *ITS*, volume 8474 of *LNCS*, pages 134–143. Springer, 2014.

[4] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, Oct 2010.

[5] Š. Pero and T. Horváth. Comparison of collaborative-filtering techniques for small-scale student performance prediction task. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*. Springer, 2015.

[6] I. Pilászy and D. Tikk. Recommending new movies: Even a few ratings are more valuable than metadata. RecSys, 2009.

[7] C. Schatten, R. Janning, and L. Schmidt-Thieme. Integration and evaluation of a machine learning sequencer in large commercial its. In *AAAI*. Springer, 2015.

[8] C. Schatten, M. Mavrikis, R. Janning, and L. Schmidt-Thieme. Matrix factorization feasibility for sequencing and adaptive support in its. In *EDM*, 2014.

[9] C. Schatten and L. Schmidt-Thieme. Adaptive content sequencing without domain information. In *CSEDU*, 2014.

[10] N. Thai-Nghe, L. Drumond, T. Horvath, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme. Factorization techniques for predicting student performance. *IGI Global*, 2011.

Towards Understanding How to Leverage Sense-making, Induction and Refinement, and Fluency to Improve Robust Learning

Shayan Doroudi[1], Kenneth Holstein[2], Vincent Aleven[2], Emma Brunskill[1]
[1]Computer Science Department, [2]Human-Computer Interaction Institute
Carnegie Mellon University
{shayand, aleven, ebrun}@cs.cmu.edu, kenneth.holstein@gmail.com

ABSTRACT

The field of EDM has focused more on modeling student knowledge than on investigating what sequences of different activity types achieve good learning outcomes. In this paper we consider three activity types, targeting sense-making, induction and refinement, and fluency building. We investigate what mix of the three types might be most effective in supporting robust student learning. To do so, we collected data from students in grades 4 and 5 who completed sequences of activities in largely random order. Students significantly improved from pretest to posttest, suggesting that incorporating all three types can support learning gains. Using hierarchical linear modeling, we found that students who get relatively more fluency problems achieve higher posttest scores. This finding suggests that fluency-building activities are most effective in helping students learn, although our data do not allow us to conclude that fluency alone is sufficient. This work represents a step towards better understanding what combination of different learning mechanisms may best support robust learning.

1. INTRODUCTION

Intelligent tutoring systems (ITSs) have been very effective at enhancing student learning [12, 6]. They typically provide step-level support for complex problem solving such as correctness feedback, next-step hints, and error-specific feedback. ITSs also provide individualized problem selection [11, 3]. It is interesting to consider ITS effectiveness from the perspective of the Knowledge-Learning-Instruction (KLI) framework [5]. KLI posits that three mechanisms of learning—sense-making (SM), induction and refinement (IR), and fluency-building processes—may all be important for robust learning (persistent learning that supports future learning) in any complex domain. However, existing ITSs typically focus only on the IR mechanism through the provision of scaffolded, tutored problem solving. It is possible that providing support for all three learning mechanisms will lead to more robust learning. Supporting the three learning mechanisms would however require a wider range of activity types than typical ITSs offer, to add or enhance support for SM and fluency. Further, it would require that we answer key questions of how and when to provide the different activity types to different learners in an individualized manner, which may itself depend on the student's learning process so far.

In this paper we take a preliminary step towards answering these questions. Fractions Tutor [8] is a web-based intelligent tutoring system for fourth and fifth grade fractions learning. We significantly extended the Fractions Tutor to support all three learning mechanisms. We then collected data from over 600 students with constrained random problem sequences. This allowed us to do a preliminary analysis to understand the contributions of activities targeting the three different learning mechanisms. We did this by fitting a hierarchical linear model (HLM) to our data to see how posttest scores are influenced by the proportion of each activity type in problem sequences as well as looking at the correlation of each activity type with posttest scores. A challenge in drawing conclusions from our data is that the mix of activity types each student was presented with was correlated with the number of problems each student did, but despite this challenge, we show that fluency-building activities are more effective for robust learning.

There has been related work on how to combine two different types of activities, such as worked examples and problem-solving practice [10]. More recent work on MOOCs has analyzed the effectiveness of different activity types chosen by the student (instead of the tutor) [4, 2]. More relevant to the current work is prior work on SM and fluency processes in the Fractions Tutor [8, 9]. While that work also uses hierarchical linear modeling [9], their model includes predictors corresponding to experimental conditions, whereas we have random trajectories with no experimental conditions. Using random sequences gives us the potential to compare a wider variety of relative compositions and sequences of activity types than a standard experimental study.

Finally, prior EDM work has looked at the related problem of how to measure the relative efficacy of different activities [1, 7]. While these works deal with a very similar problem to ours, they differ in at least two main respects from the present work. First, their models consider the efficacy of different activities in performance while being tutored, whereas

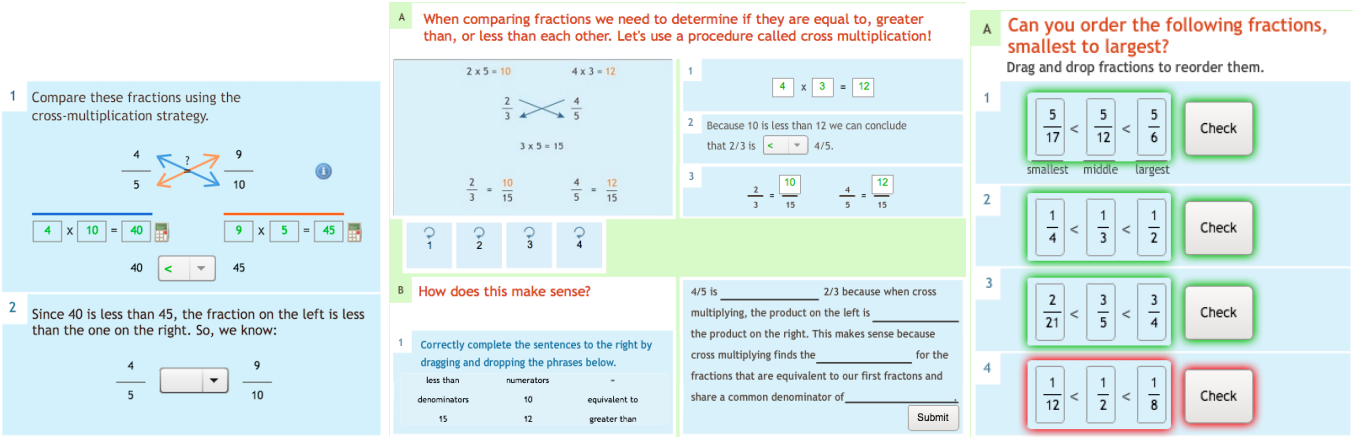


Figure 1: Sample IR (left), SM (center), and fluency (right) activities.

we are interested in robust learning (i.e. performance on a posttest). Second, they only consider which individual activity is best rather than what mix of activities is best. Our modeling approach could in theory suggest optimal mixes of activity types, although we find that in this case the best fitting model reduces to one that can only suggest the relative efficacy of each activity type. It would be worthwhile to compare our findings with the results we can obtain from these models as next steps of our work.

2. METHODS

2.1 Fractions Tutor

For this work, our Fractions Tutor covered topics emphasized in the Common Core[1]: making and naming fractions, fraction equivalence and comparison, and fraction addition. For each topic, we designed three activity types designed to promote each of the KLI learning mechanisms. KLI does not provide strict design guidelines and so we now describe how our designed activities targeting each learning mechanism are in line with KLI's definitions.

Under KLI, IR processes are non-verbal learning processes that improve the accuracy of knowledge [5]. Activities to promote IR processes emphasized procedural learning and practice via fine-grained task decomposition and step-level guidance and feedback, as is typical of ITSs [11]. An IR activity for a procedure for the comparison of two fractions is shown in Figure 1, on the left.

In KLI, SM processes are "explicit, verbally mediated learning in which students attempt to understand or reason" [5]. Our SM activities included instructional videos designed to promote conceptual understanding of targeted fractions topics. The videos were divided into small segments and interspersed with brief supporting problem-solving exercises. Each SM activity concluded with a drag-and-drop fill-in-the-blank question designed to help students self-explain the underlying concepts. An example SM activity for the cross-multiplication procedure is shown in Figure 1 (center). Unlike the IR activities that teach the application of this procedure, the SM activities were designed to help students understand why a certain procedure (e.g., cross-multiplication to compare and order fractions) is effective.

Finally, under KLI, fluency-building processes are non-verbal processes that strengthen memory and enable students to apply their procedural knowledge faster and more fluently [5]. Thus the fluency activities were designed to promote the development of rapid reasoning about fractions and fluent performance on minimally-decomposed problem-solving exercises. Whereas students received support from the tutor via step-level hints in IR activities and video-replays in SM activities, neither were available in fluency activities. See a sample fluency activity in Figure 1, on the right.

2.2 Activity Selection

Since we wish to be able to understand a broader range of activity orderings and mixes rather than a small fixed set, we presented activities to students in a semi-randomized order. A semi-randomized order was chosen as a compromise between two potentially competing objectives. The first is to enhance student learning broadly and for the students that participated in this initial data collection. This objective would push us towards selecting an activity order that draws upon existing research on effective sequencing and satisfies commonly assumed topic orderings. Our second objective is to be able to find effective (potentially adaptive) orderings that may fall outside of the reach of standard procedures. To balance these two competing objectives, we chose to provide students with activity sequences that initially satisfy a prerequisite structure over activity types and topics (designed by the authors). Students could be presented with any activity whose prerequisites had already been presented. This ensured some semantic ordering, e.g. students would not be presented with addition problems before being introduced to the concept of a fraction! However, only a fixed set of 26 problems have prerequisites; once a student finishes the first 26 problems, the student is randomly presented with problems from a large pool of remaining problems.

[1]The Common Core State Standards determine the math curriculum for students from kindergarten through high school in most US states: http://www.corestandards.org/.

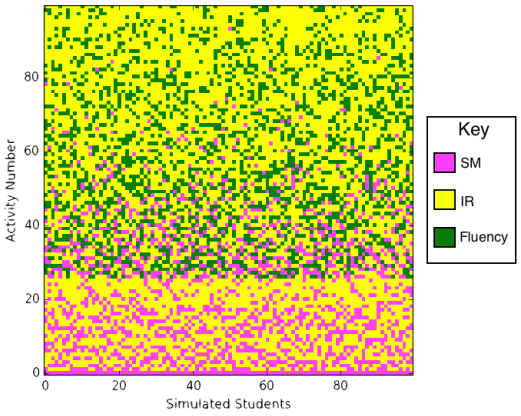


Figure 2: Simulation of potential activity type orderings. Each column represents a sequence of activity types for a student who was given 100 problems.

2.3 Data Collection

We collected data from students using the tutor in eight schools spanning two school districts. Students took a pretest, used the tutor for several sessions, and then took a posttest. The pretest and posttest consisted of 16 items covering conceptual and procedural understanding over skills involved with the three topics. Items were developed by building off of Common Core standards and prior assessment items developed for the Fractions Tutor. For our data analysis we used data from students who started each of the pretest and posttest (639 students).

3. ANALYSIS AND RESULTS

Our ultimate objective for this initial analysis was (1) to evaluate if the new tutor helped students improve their understanding of the material, and (2) to determine what static mix of activity types (SM, IR, and fluency) has the most effective learning outcomes.

3.1 Learning Gains

The mean pretest score is 5.82 ± 3.19 and the mean post test score is 8.23 ± 2.78 (both out of 16). Students significantly improved from pretest to posttest (paired t-test, $t(638) = 27.67$, $p < 10^{-110}$). The effect size was $d = 1.09$, which is considered a large effect size. These results demonstrate that our assortment of activity types can support learning gains, even when those activities are largely randomized.

3.2 Correlation of Variables

Exploratory data analysis revealed a substantial variation in both the number of activities done (mean: 49.6 ± 30.9) and the amount of time students had with the tutor (mean: 183.2 ± 82.3 minutes). Due to the prerequisite structure and semi-randomized ordering used, the number of activities and amount of time spent on the tutor influenced the relative proportion of each activity type that the students completed. To see this we can look at a set of possible simulated sequences that could have been given to students: Figure 2 shows 100 such sequences of 100 problems each. We can observe that students completing 26 problems or less would only receive SM and IR problems. In addition, because the total number of SM activities was fewer than the other two types of activities, if a student did a very large number of activities, the fraction of activities he/she completed would eventually be dominated by IR and fluency.

Predictor	Pearson's r	Partial Pearson's r	p-value
SM	-0.48	-0.15	$5.8 * 10^{-4}$
IR	0.26	-0.033	1
Fluency	0.44	0.18	$5.0 * 10^{-6}$

Table 1: Pearson's r between proportion of problem types and posttest scores, along with partial correlation coefficients when controlling for the number of problems done and amount of time spent on the tutor and Bonferroni corrected p-values for the partial correlations. Predictor variables represent the proportion of problems done by the student that were SM, IR, or F.

To help tease apart the strong correlation between the number of problems and the distribution of activity types completed, we computed the partial correlation between the proportion of problems belonging to each activity type and the posttest score, controlling for both the total number of problems done as well as the amount of time spent by the student. The results are shown in Table 1.

The decrease in magnitude between the raw correlation and partial correlation for each activity type tells us that the number of problems done and total time spent on the tutor accounts for some of the correlation with post test, as expected. More interestingly, the proportion of fluency problems is significantly positively correlated with the posttest scores even after considering the number of problems done and time spent. This suggests that having relatively more fluency problems is beneficial for students, beyond the fact that the students who did more fluency problems tend to have completed more problems; we will verify this with our hierarchical linear modeling. On the other hand, the proportion of SM problems is significantly negatively correlated with the posttest score even after accounting for time and number of problems.

To limit the extent to which students who got more time tended towards a certain mix of activity types, we restricted our subsequent analysis to only those students from one school district who had 150-200 minutes of tutor time in between pretest and posttest (resulting in 268 students).

3.3 Impact of Activity Proportions

The second key issue we wished to investigate was how student learning may be influenced by the mix of different activity types that they complete. To address this issue, we used hierarchical linear modeling to predict posttest scores as a function of the mix of SM, IR and fluency problems that a student completed. In the analysis below, we consider two-level HLMs that treat the class the student is from as a level-2 variable. Using a two-level model resulted in a better fit than just using linear regression. (We tried adding school as a level-3 variable, but this did not improve the

Predictor	Coefficient	p-value
Intercept	12.97	$5.4 * 10^{-9}$
Pretest Score	0.59	$< 1.0 * 10^{-15}$
Proportion SM	-11.20	$9.0 * 10^{-8}$
Proportion IR	-7.67	.021

Table 2: The coefficients of the HLM and their significance with a Bonferroni correction for doing four t-tests. (Satterthwaite approximations were used to compute the degrees of freedom.)

fit.) After trying a variety of models, we found that the best fitting model (in terms of cross-validated RMSE) was one of the simplest. The best model used only three predictor variables: pretest score, proportion of SM problems, and proportion IR problems. (Note that proportion of fluency problems is not a necessary predictor since the three proportions sum to one.) The coefficients for the level-1 variables of the HLM and their p-values are given in Table 2. We see the coefficient for the proportion of SM and the coefficient for the proportion of IR were significant and negative. Thus our model suggests fluency is the most effective activity type (since minimizing the proportion of IR and SM maximizes the posttest score) followed by IR, which agrees with our partial correlation analysis. The apparent lack of efficacy of SM problems may be because these items were substantially more time consuming for students to complete than the two other activity types. Thus even if SM problems are useful, their relative effectiveness per time spent may be lower than more active problems. This is also supported by recent results on the benefit of learning by doing [4].

If our model generalized to all possible sequences, it would suggest that students should do as many fluency problems as possible and not do any SM or IR problems. To allow for non-trivial mixes of activity types, the model would need to include interaction terms between the proportions of different activity types. Such models had statistically insignificant coefficients and worse fits than the model presented.

Nonetheless, it is important to note that any student who did fluency problems in our study necessarily also did SM and IR problems due to the prerequisite structure. Therefore we cannot reliably evaluate the value of a sequence consisting of only a single activity type using our model; such a sequence is *very* different from sequences the students actually received. Rather, the conclusion we can draw from our model is that if we were able to provide additional tutoring to students who already did many problems using our tutor, we should probably just give them more fluency problems.

Notice that our model includes no term for the total number of problems a student did (which we know correlates well with the posttest score). When adding such a term to our model, the fit was worse and the coefficient for that term was both small and statistically insignificant. This implies that the proportion of fluency problems is a better predictor than the number of problems a student did!

4. CONCLUSION

We have extended an existing ITS to include activity types that support all three learning mechanisms posited by the Knowledge-Learning-Instruction framework. In a large-scale classroom study, our ITS had learning gains with a large effect size. A preliminary analysis indicates that students who have a high percentage of fluency problems have the largest posttest scores, suggesting that fluency-building activities are most effective in helping students learn. However, many open questions remain. To what extent are SM and IR problems necessary? Does the appropriate mix of activity types differ for different topics (e.g. making fractions vs. fractions addition)? We hope to address these questions as we work towards our goal of learning personalized policies that best support robust student learning.

5. ACKNOWLEDGEMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A130215 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept. of Education.

6. REFERENCES

[1] J. E. Beck and J. Mostow. How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In *Proc. of ITS*, pages 353–362, 2008.

[2] J. Champaign, K. F. Colvin, A. Liu, C. Fredericks, D. Seaton, and D. E. Pritchard. Correlating skill and improvement in 2 MOOCs with a student's time on tasks. In *Proc. of L@S*, pages 11–20, 2014.

[3] A. Corbett, M. McLaughlin, and K. C. Scarpinatto. modeling student knowledge: cognitive tutors in high school and college. *UMUI*, 10:81–108, 2000.

[4] K. Koedinger. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proc. of L@S*, 2015.

[5] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 4 2012.

[6] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of Cognitive Tutor Algebra I at scale. *Educational Eval. & Policy Analysis*, 2013.

[7] Z. A. Pardos and N. T. Heffernan. Detecting the learning value of items in a randomized problem set. In *Proc. of AIED*, 2009.

[8] M. A. Rau, V. Aleven, and N. Rummel. Complementary effects of sense-making and fluency-building support for connection making: A matter of sequence? In *AIED*, 2013.

[9] M. A. Rau, V. Aleven, N. Rummel, and S. Rohrbach. Sense making alone doesnt́ do it: Fluency matters too! its support for robust learning with multiple representations. In *Proc. of ITS*, pages 174–184, 2012.

[10] R. J. Salden, K. R. Koedinger, A. Renkl, V. Aleven, and B. M. McLaren. Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review*, 22(4):379–392, 2010.

[11] K. VanLehn. The behavior of tutoring systems. *IJAIED*, 16(3):227–265, 2006.

[12] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Learning Behavior Characterization with Multi-Feature, Hierarchical Activity Sequences

Cheng Ye
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
cheng.ye@vanderbilt.edu

John S. Kinnebrew
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
john.s.kinnebrew@vanderbilt.edu

James R. Segedy
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
james.segedy@vanderbilt.edu

Gautam Biswas
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
gautam.biswas@vanderbilt.edu

ABSTRACT

This paper discusses Multi-Feature Hierarchical Sequential PAttern Mining, MFH-SPAM, a novel algorithm that efficiently extracts patterns from students' learning activity sequences. This algorithm extends an existing sequential pattern mining algorithm by dynamically selecting the level of specificity for hierarchically-defined features individually for each pattern. Consequently, MFH-SPAM operates on a larger space of patterns in the activity sequences. In this paper, we employ a differential version of MFH-SPAM to extract a small set of patterns that best differentiate students with different learning behavior profiles in the Betty's Brain system. Our results illustrate that: (1) MFH-SPAM identifies important patterns missed by traditional sequence mining approaches; and (2) the differential patterns provide additional information for characterizing learning behaviors. This has implications for developing targeted and adaptive scaffolding in open-ended learning environments.

1. INTRODUCTION

Open-Ended Learning Environments (OELEs [4, 7]) present students with a challenging problem-solving task, along with resources and tools for solving the task. Students have the choice to explore, and, therefore, can evolve their solutions in a variety of ways. In previous work, we proposed a theory-based approach called *coherence analysis* (CA) [7] for analyzing student behavior in OELEs. Experimental results showed that grouping students using the CA metrics produced distinct behavior profiles that are discussed in greater detail in Sections 3 and 4. To date we have established the stability and usefulness of our CA measures across extended periods of student work, which does not make this approach directly applicable to adaptive scaffolding as students work in the OELE. To address this problem, our goal has been to use sequence mining methods to find students' activity patterns that are indicators of their behavior profiles. In this paper, we present a case study illustrating that action patterns derived using a novel hierarchical sequence mining approach followed by differential analysis enable classification performance on a par with the groupings derived using CA. Occurrence of individual action patterns can be easily detected online, and future work will assess their utility for early identification of behavior profiles and contextualized scaffolding in OELEs.

In the Betty's Brain OELE [5] each action performed by a student has a number of accompanying features that capture context and consequences of the action. In past work, we used pre-processing methods to select specific features and the level of granularity for each feature to generate 'flat' sequences for pattern mining [2]. This largely ad hoc process resulted in our running many different mining analyses, but often missing potentially important patterns. Other work, such as Plantevit et al. [6], has addressed some aspects of the search in large feature spaces. They define a two-phase technique that first determines frequent combinations of features and levels of specificity in hierarchical representations to pre-processes multi-feature (hierarchical) sequences into a 'flattened' representation. While this approach provides clear advantages over numerous mining analyses with ad hoc feature and granularity choices, many frequent patterns can still be missed due to the initial flattening phase.To address this issue, we have developed a novel **M**ulti-**F**eature, **H**ierarchical **S**equential **PA**ttern **M**ining algorithm (MFH-SPAM).

MFH-SPAM extends the sequence mining algorithm SPAM [1] to simultaneously operate on the entire feature space of action sequences for pattern mining. In this work, we start with MFH-SPAM, and then apply a classifier wrapper method [3] to discover a small subset of mined patterns that are useful for differentiating students across the CA-derived learn-

ing behavior profiles. We have evaluated MFH-SPAM and other traditional sequence mining approaches in this behavior profile classification task using data from a recent study with the *Betty's Brain* OELE. Results show that MFH-SPAM consistently outperforms traditional sequence mining approaches on this task. Further, the differential patterns provide additional information for characterizing student learning behaviors, which has implications for developing targeted and adaptive scaffolding in OELEs.

2. MFH-SPAM APPROACH

Our approach to efficient mining of Multi-Feature, Hierarchical (MFH) sequences extends the SPAM algorithm [1] by directly working with the MFH representation of actions during the mining process. To illustrate this representation, we consider a generic set of possible items/actions to make up sequences (A, B, or C) with an additional feature (*e.g.*, a measure of the action's outcome) that can take on values of + or − at the most general level. In this example, + values for the outcome feature can be further specified as either $+Big$ or $+Small$ at the next level of the hierarchy. Therefore, an individual action might be represented as $B^{+\text{Big}}$, and both $B^{+\text{Big}}$ and $B^{+\text{Small}}$ actions could be more generally represented as B^{+} by abstracting the outcome feature to the more general + level. Further, $B^{+\text{Big}}$, $B^{+\text{Small}}$, and B^{-} actions could all be represented as simply a B action by ignoring the outcome feature entirely. We represent one action followed by another in a sequential pattern using the $\rightarrow$ symbol, such as $A \rightarrow B$ to indicate A followed by B. Itemsets (*i.e.*, co-occurring items in the sequence) are surrounded with parentheses, such as (A, B) to indicate both A and B occurring at the same position in a sequence (*i.e.*, simultaneously).

The core SPAM [1] algorithm searches the space of possible sequential patterns by incrementally extending the current pattern (starting with an empty pattern) in a depth-first manner. For each pattern in the search, SPAM generates the potential "child" patterns by applying one of two types of extensions to the current pattern: 1) a Sequence-extension step (S-step), which appends an item to the end of the sequence (occurring after the last item/itemset), or 2) an Itemset-extension step (I-step), which adds an additional item to the last itemset in the current pattern. For each pattern considered, SPAM calculates the number of sequences in which the pattern occurs using a vertical bitmap representation, explained in more detail later. If the number of sequences in which the new pattern is contained is less than the specified support threshold, SPAM rejects the pattern and does not consider any subsequent extensions to it.

MFH-SPAM augments SPAM with two new pattern extension steps in the pattern search: Feature extensions (F-steps) and Hierarchical extensions (H-steps). During an F-step, MFH-SPAM adds an additional feature to the last item of the current sequence using one of the most general values in the feature hierarchy. For example, the possible extensions to the pattern $A \rightarrow B$ with an F-step would result in $A \rightarrow B^{+}$ or $A \rightarrow B^{-}$. During an H-step, MFH-SPAM selects the last feature of the last item of the current sequence and specifies its value at one level deeper in the feature hierarchy. For example, the possible extensions to the pattern $A \rightarrow B^{+}$ with an H-step would result in $A \rightarrow B^{+\text{Big}}$ or $A \rightarrow B^{+\text{Small}}$.

In addition to these two new extension steps in MFH-SPAM, we define a corresponding extension to the vertical bitmap approach employed in SPAM to efficiently calculate the support for a new pattern[1]. For each data sequence, SPAM initially defines a bitmap for each possible item (*e.g.*, A, B, and C) that represents the locations of that item in the sequence with a value of 1 (all other locations have a value of 0). For example, the sequence $A \rightarrow B \rightarrow B$ would be represented with an A bitmap of [1 0 0], a B bitmap of [0 1 1], and a C bitmap of [0 0 0]. As SPAM generates patterns, it combines item bitmaps to produce *pattern bitmaps* in which 1's represent the endpoints of the corresponding pattern in the sequences. Consequently, for a trivial, single-item pattern like A, the pattern bitmap is exactly the same as the initial item bitmap.

For an S-step extension of a pattern (*e.g.*, extending A to $A \rightarrow B$), SPAM first transforms the current pattern bitmap ([1 0 0]) to indicate where the extension to the current pattern could occur. This is performed by shifting the bitmap to make each location following the occurrence of a 1 in the pattern bitmap a 1 (indicating a *candidate location* for the additional item being added in the S-step) and making all other locations 0 (e.g., resulting in the bitmap [0 1 0]). In other words, $A \rightarrow B$ exists in the sequence if B exists in the candidate location of the second position in the sequence. To complete the S-step (*e.g.*, for A to $A \rightarrow B$) SPAM performs a bitwise AND operation on the transformed pattern bitmap and the item (B) bitmap, resulting in the new pattern bitmap of [0 1 0] indicating that the pattern $A \rightarrow B$ exists and ends at the second position in the sequence.

We extend the SPAM bitmap procedure in F- and H-steps by first creating bitmaps for each possible feature value (at every level of the hierarchy) in the sequence, just as SPAM does with each possible item. Thus, if the original sequence were $A^{-} \rightarrow B^{+\text{Big}} \rightarrow B^{+\text{Small}}$, we would have a − bitmap of [1 0 0], a + bitmap of [0 1 1], a $+Big$ bitmap of [0 1 0], and a $+Small$ bitmap of [0 0 1]. The bitmap operations for F- and H-steps are then analogous to those for S-steps except without the bitmap shift[2] and using the feature value bitmap corresponding to the chosen extension. For example, applying an F-step to add the outcome feature with a value of + to the pattern $A \rightarrow B$, producing $A \rightarrow B^{+}$, would correspond to [0 1 0] (the pattern bitmap) AND [0 1 1] (the feature value bitmap), giving the new pattern bitmap [0 1 0], indicating that this pattern does occur in the example sequence and ends at the second position in the sequence. With the additional F- and H-steps, as well as corresponding bitmap operations for calculating support, MFH-SPAM extends SPAM to efficiently search the space of possible patterns in MFH sequences. Finally, to choose a small subset of the frequent patterns identified by MFH-SPAM (or by SPAM for the experimental comparison) that differentiate the pre-defined learning profiles, we apply a classifier wrap-

[1]In the algorithm description, we describe only the case in which no gaps are allowed between items in the pattern, however, implementing more general gap constraints works in the same manner as with extensions to the original SPAM algorithm

[2]No shift is necessary because the candidate location is for adding further detail to the last item in the current pattern rather than adding an item after it.

per method [3]. Using a greedy approach, the classifier wrapper iteratively identifies the best pattern to include next[3].

3. DATA AND EVALUATION METHODS

The data presented in this paper comes from a study of 98 students from four middle school science classrooms using *Betty's Brain* for six weeks [7]. Six coherence measures were employed to describe the quality and quantity of various problem-solving activities for each student, and hierarchical clustering with these measures identified three primary clusters of students characterized by different behavior profiles [7]. In total, 87 of the students fell into one of these three clusters, and the other 11 students exhibited behavior profiles indicative of either extreme confusion or disengagement. The primary clusters were defined as:(1) *Frequent researchers and careful editors*, who spent large proportions of their time viewing sources of information and did not edit their maps very often; (2) *Strategic experimenters*, who spent a fair proportion of their time viewing sources of information, but often did not take advantage of this information; and (3) *Engaged & efficient students*, who edited their maps very frequently, and usually supported by information from previous activities.

To generate MFH activity sequences for mining, we categorized learning actions into seven primary categories, defined hierarchically (these categories are discussed in more detail in [2]): *Read*ing resource pages; *Search*ing the resources for keywords; causal *Map Edit*ing; *Query*ing the teachable agent, Betty; having Betty take a *Quiz*; asking Betty to *Explain* her answer; or taking *Note*s or causal link annotations (*LinkEval*) indicating whether a link is believed to be correct. To capture the context associated with these actions, we use additional features: (1) the "Length" dimension (applied to Read actions) indicates whether the student spent enough time on the page to have read a significant amount of the material (Full) or only spent a brief period of time on the page (Short) [2]; (2) the "Previous (Full) Read" dimension indicates whether the student has previously done an in-depth ("Full") read of the page or not; (3) the "Supported" dimension indicates whether or not an EditLink action was based on either recently viewed reading materials or quiz results [7], with supported actions denoted by *Sup* and unsupported actions denoted by *NoSup*; and (4) the "Map Score Change" dimension indicates what effect an EditLink action had on the quality of the student's map - whether the quality improved (denoted by +), worsened (denoted by −), or did not change (denoted by =).

We evaluate our MFH-SPAM approach with comparison to four alternative approaches: *Flattened Features (SPAM)* first flattened all activity sequences using all features and the greatest level of action specificity and then used SPAM to generate candidate patterns (e.g., this approach would consider the pattern $\text{LinkRem}^{+}{}_{\text{Sup}} \rightarrow \text{LinkAdd}^{-}{}_{\text{NoSup}}$, but it would not consider the more general pattern LinkEdit → LinkEdit); *Actions-only (SPAM)* considered only the frequent patterns at the most general level of specificity and did not consider any additional features; *MFH-SPAM Baseline by Frequency* used our MFH-SPAM algorithm to generate candidate patterns and simply selected the 10 most frequent patterns; and *Coherence Metrics* classified students using the coherence measures. The performance of each approach was evaluated as the average F1 score of the resulting classifier using 10-fold cross-validation. We chose decision trees as the classifiers and performed this analysis at mining support thresholds ranging from 1.0 to 0.5 in increments of 0.02.

[3]A limit of 10 patterns and an increase of at least 0.1% in performance over the previous pattern set was used in our implementation of the wrapper. A stratified 5-fold cross-validation approach was used for building the classifier in the wrapper with F1 score for evaluation.

4. RESULTS

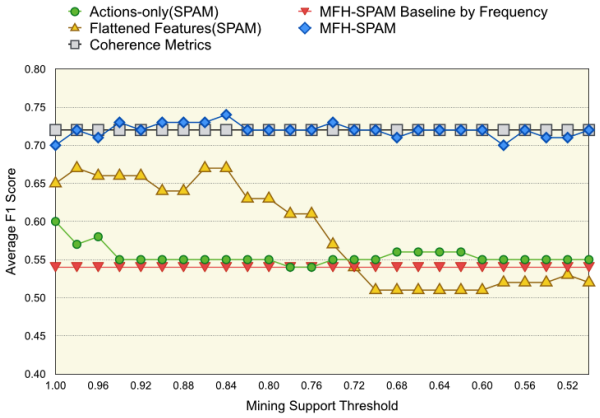


Figure 1: Classification performances of MFH-SPAM and alternative approaches

Figure 1 illustrates the performances of the classifiers built using the candidate feature sets mined in each approach. At each level of support, MFH-SPAM achieved an average F1 score that was much higher than the scores produced by the other sequence mining methods. When using a particularly high mining support threshold, the Flattened Features approach achieves performance close to that of MFH-SPAM, but its performance decreases dramatically as the support threshold is reduced (and the search space is increased). One striking result from this analysis is that MFH-SPAM's performance is on par with the performance of the classifier trained with the features used to perform the original clustering that defined these behavior profile classes. Further, Table 1 presents the five patterns chosen most frequently across the 10 cross-validation folds at a support threshold of 0.9. Considering these top patterns, it is clear that the first three patterns could not have been identified without MFH-SPAM, as they involve multiple levels of hierarchies and feature specificity.

Interestingly, the top MFH-SPAM patterns all involve various forms of causal link edits. This suggests that the way a student went about building their map, as opposed to the way they navigated the resources and investigated Betty's quiz results, was the most useful in predicting their overall learning behavior profile. However, the edit actions, through the support feature, can also incorporate the action's relationship to reading and quiz actions. In other words, what was most helpful in predicting a student's cluster was not the way they acquired information (either from the resources or quiz results), but how they applied previously acquired information to editing their maps. When comparing frequency of use across the three groups, their relative magnitudes are

Table 1: Pattern Frequency Mean (Std Dev) by Cluster for MFH Wrapper with Support 0.9

Pattern	Researchers	Experimenters	Efficient
$\text{LinkRem}^{+}{}_{\text{Sup}} \rightarrow \text{LinkEdit}$	2.6 (2.5)	3.6 (3.0)	14.3 (8.4)
$\text{LinkEdit}^{+}{}_{\text{Sup}} \rightarrow \text{LinkAdd}$	2.3 (1.9)	2.5 (2.6)	12.0 (6.5)
$\text{LinkEdit}^{+}{}_{\text{NoSup}} \rightarrow \text{LinkEdit}$	3.3 (2.9)	16.4 (16.9)	15.6 (12.3)
$\text{LinkEdit}^{-} \rightarrow \text{LinkEdit}^{-}$	3.7 (3.1)	17.5 (16.0)	18.3 (16.2)
LinkAdd^{-}	15.3 (7.2)	28.6 (12.1)	43.5 (21.6)

compatible with the behavior descriptions; *e.g.*, researchers and careful editors make the least number of these edits; engaged & efficient students have the most; and strategic experimenters fall in between. This confirms that the engaged & efficient students, who exhibited the best learning behaviors and the largest learning gains [7], are broadly distinguished from the other groups by more map editing overall: ineffective and effective; supported and unsupported. The usage distributions for these patterns also revealed interesting characteristics about strategic experimenters. These students performed patterns with supported edits far less frequently than engaged & efficient students. Conversely, they performed patterns with unsupported edits far more frequently than researchers and careful editors. Thus, even though the engaged & efficient students made several unsupported and ineffective edits, it would seem that their overall edit distribution is far more favorable to achieving better map scores (and in their case, better pre-post gains on domain knowledge) than that of the strategic experimenters.

To better characterize these three groups, we followed up on previous experimental results [2] and further analyzed the top behavior pattern: (1) $LinkRem^{+}{}_{\text{Sup}} \rightarrow LinkEdit$ that indicates an effective map correction behavior (removing an incorrect link with supporting evidence) followed by further editing. Overall, an average of 19% (s.d. 9%) of the engaged/efficient students' total number of link edits involved this pattern versus 9% (s.d. 8%) for researchers/careful-editors and 9% (s.d. 7%) for strategic experiments. This behavior of incorporating effective map correction in periods of extended map editing appears to be a key characteristic of the engaged/efficient students. Further analysis also suggested that engaged/efficient students were relatively more likely to follow this pattern with a quiz to evaluate their revised map than the researchers/careful-editors and strategic experimenters. This may indicate a greater propensity for the engaged/efficient students to effectively combine evaluation of the causal map with map construction and correction. In summary, going back to OELE characteristics, the engaged and efficient students seem to be better at exploring the problem-solving space, and in distinguishing correct and incorrect approaches to solving complex problems.

5. DISCUSSION AND CONCLUSIONS

MFH-SPAM provides a comprehensive approach to mining OELE activity sequences by efficiently covering the entire MFH action-feature space to generate patterns. Results showed that MFH-SPAM consistently outperforms traditional sequence mining approaches on a behavior profile classification task. Further, analysis of the MFH-SPAM patterns illustrated that a nice, compact way for differentiating these student groups, while retaining high accuracy, was in their approach to map construction and refinement using various forms of editing actions. Overall, these results showed the importance of behavior patterns identified by MFH-SPAM and illustrated the potential to use these patterns to better characterize and ultimately scaffold student learning. In general, effective virtual agents for adaptive scaffolding in OELEs like Betty's Brain may do well to focus on behavior patterns to gain an understanding of how students' *apply their acquired knowledge* (*e.g.*, from reading the resources and studying quiz results) to build and refine models. Detection of specific suboptimal (not using acquired information well) or erroneous behaviors in this context may provide the needed cue for effective scaffolding.

6. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A120186 to Vanderbilt University. The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

7. REFERENCES

[1] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 429–435. ACM, 2002.

[2] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1):190–219, 2013.

[3] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[4] S. Land. Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3):61–78, 2000.

[5] K. Leelawong and G. Biswas. Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[6] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong. Mining multidimensional and multilevel sequential patterns. *ACM Transactions on Knowledge Discovery from Data*, 4(1):4:1–4:37, Jan. 2010.

[7] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, To Appear.

Discrimination-Aware Classifiers for Student Performance Prediction

Ling Luo
School of Information Technologies, The University of Sydney, Australia
National ICT Australia
ling.luo@sydney.edu.au

Irena Koprinska
School of Information Technologies, The University of Sydney, Australia
irena.koprinska@sydney.edu.au

Wei Liu
Faculty of Engineering & IT, University of Technology Sydney, Australia
National ICT Australia
wei.liu@uts.edu.au

ABSTRACT

In this paper we consider discrimination-aware classification of educational data. Mining and using rules that distinguish groups of students based on sensitive attributes such as gender and nationality may lead to discrimination. It is desirable to keep the sensitive attributes during the training of a classifier to avoid information loss but decrease the undesirable correlation between the sensitive attributes and the class attribute when building the classifier. We illustrate, motivate, and solve the problem, and present a case study for predicting student exam performance based on enrolment information and assessment results during the semester. We evaluate the performance of two discrimination-aware classifiers and compare them with their non-discrimination-aware counterparts. The results show that the discrimination-aware classifiers are able to reduce discrimination with trivial loss in accuracy. The proposed method can help teachers to predict student performance accurately without discrimination.

Keywords

Predicting student performance; association rule mining; decision tree; discrimination-aware classification

1. INTRODUCTION

Educational data often contains sensitive attributes such as age, gender and nationality. Mining such data may generate discriminating rules. For example, if our goal is to predict the exam mark of current students, and in the historic dataset used for training of the prediction algorithm, males have achieved significantly higher exam marks than females, a prediction rule using the attribute gender may be generated. It may produce high accuracy but we cannot use it for providing feedback to students or other decision making, as it can be seen as discriminating based on gender, which is unethical and also against the law. Sensitive attributes such as gender should be used as an information carrier and not as distinguishing factors [1]. In this paper we consider building discrimination-aware classification models for predicting student performance.

The task of discrimination-aware classification can be defined as follows [2; 3]: given a labelled dataset and an attribute *S*, find a classifier with high accuracy that does not discriminate on the basis of *S*. There are two approaches to deal with this problem: 1) not using the sensitive attribute to build the classifier and 2) modifying the classification algorithm by integrating a discrimination-aware mechanism to reduce discrimination. The first approach, simply removing the sensitive attribute from the training data, results in information loss and also typically doesn't solve the problem as other attributes are correlated with the sensitive attribute, and will discriminate indirectly. In this paper, we develop and apply methods from the second group which incorporate discrimination awareness during the building of the classifier and use information from the sensitive attribute without causing discrimination.

There are two important aspects that need to be considered when applying discrimination-aware classifiers in educational settings. Firstly, adjusting the classifier to reduce discrimination typically leads to lower predictive accuracy. Given this trade-off between accuracy and discrimination, our aim is to build a classifier with lower discrimination without significant loss in accuracy. Secondly, the output of the classifier should be easy to understand and use by teachers and students. Therefore, we consider classifiers based on decision tree and association rules, which generate sets of rules to guide prediction and decision making.

Our contribution can be summarized as follows:

- We illustrate and motivate the problem of discrimination-aware classification for mining educational data, and show its importance and challenges in educational data mining. Discrimination-aware classification has not been studied for educational data mining and our main goal is to raise the awareness of the community to this problem.
- We introduce our recently proposed classification method Discrimination-aware Association Rule classifier (DAAR) [4]. DAAR uses the novel Discrimination Correlation Indicator (DCI) to measure the discrimination severity of an association rule and select non-discriminatory rules.
- We consider the task of predicting the student exam performance in a first year computer programming course. We apply two discrimination-aware classifiers: our method DAAR and the state-of-the-art Discrimination-Aware Decision Tree (DADT) [3], and compare their performance with standard non-discrimination-aware association rules and decision tree. We show that both DAAR and DADT are able to produce non-discriminatory rules with minimum loss in accuracy.

2. RELATED WORK

Mining educational data to predict student performance has gained increasing popularity. Romero et al. [5] predicted the final student mark based on the Moodle usage data such as the number of passed and failed quizzes, number of completed assignments, number of sent and read messages on the discussion board and the time spent on the assignments, quizzes and discussion board. In their subsequent work [6], the same group studied predicting the

student grade (pass or fail) based on the student participation in a discussion forum, using a number of machine learning algorithms, in the middle and at the end of the semester. Kotsiantis et al. [7] applied an ensemble of classifiers to predict the exam grade (pass or fail) from assessment data during the semester in an online informatics course. Lykourentzou et al. [8] predicted dropouts and completers in e-learning courses on computer networks and web design, using demographic and assessment data.

The discrimination-aware classification problem was introduced in by Pedreshi et al. [2] and Kamiran and Calders [9]. Discrimination-aware naïve Bayes approaches were proposed in [1] and discrimination-aware decision trees were developed in [3].

In this paper, we investigate discrimination-aware classifiers for mining of educational data. We apply our recently proposed discrimination-aware classifier based on association rules and also a discrimination-aware decision tree. We show how these algorithms can be applied for predicting student performance in a first year programming course, discuss the results, and raise the awareness of the Educational Data Mining community to the importance of discrimination-free classification.

3. METHODOLOGY

In this section we describe the main principles of the two discrimination-aware classifiers: our method DAAR and the state-of-the-art DADT. Both classifiers are designed to decrease the discrimination of the predictive model with minimal impact on the accuracy. They are based on the popular and successful association rule classifiers and decision trees, which produce rules that can be easily understood and directly applied by teachers and students.

3.1 Association Rule Classifiers and DAAR

Association analysis discovers relationships among items in a dataset. An association rule takes the form $X \rightarrow Y$, where X and Y are disjoint item sets [10]. Two measures, *support* and *confidence*, are used to evaluate the quality of an association rule. Given a dataset containing N instances and an association rule $X \rightarrow Y$, the support and confidence of this rule are defined as:

$$\text{Support}\ (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}, \qquad \text{Confidence}\ (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

where $\sigma(\cdot)$ is the frequency of an item set $(\cdot)$. High-quality rules have high support and confidence.

Classification Based on Association (CBA) [10] uses association rules to solve classification problems. In a standard association rule, any attribute which is not included in X, can appear in Y while in CBA only class attributes can appear in Y.

3.1.1 DCI Measure

To measure the degree of discrimination for an association rule, we propose a new measure called DCI. Given a rule $X \rightarrow y$ and a sensitive attribute S, DCI is defined as:

$$\text{DCI} = \begin{cases} \dfrac{|\text{P}(\text{C} = \text{y}|\ \text{S} = \ \text{S}_{\text{rule}}) - \text{P}(\text{C} = \text{y}|\ \text{S} = \ \text{S}_{\text{others}})|}{\left(\text{P}(\text{C} = \text{y}|\ \text{S} = \ \text{S}_{\text{rule}}) + \text{P}(\text{C} = \text{y}|\ \text{S} = \ \text{S}_{\text{others}})\right)} \\ 0 \qquad \text{if either of the above P}(\cdot) \text{ is 0} \end{cases}$$

where $P(C = y|S = S_{\text{rule}})$ is the probability of the class to be y given the value of the sensitive attribute S is S_{rule}.

When S is a binary or multi-valued attribute, the specific S value in the rule is considered as S_{rule}, and the S_{others} includes the set of all attribute values except the one which appears in the rule. For example, if the rule is "*gender = female, degree = CS → assessment = low*", where *gender* is the sensitive attribute, then S_{rule} refers to *female*, and S_{others} refers to *male*. The DCI for this rule will be:

$$\frac{|\text{P}(\text{C} = \text{low}|\text{gender} = \text{female}) - \text{P}(\text{C} = \text{low}|\text{gender} = \text{male})|}{\text{P}(\text{C} = \text{low}|\text{gender} = \text{female}) + \text{P}(\text{C} = \text{low}|\text{gender} = \text{male})}$$

When the sensitive attribute does not appear in that rule, we define DCI to be 0.

Therefore, DCI has a range of [0, 1) and its interpretation is the following:

- If DCI is 0, the rule is free of discrimination. DCI is 0 when the probability of the class value to be y is the same for different values of the sensitive attribute S.
- If DCI is not 0, the higher the value, the more discriminatory the rule is with respect to the sensitive attribute S. Thus, the DCI value is monotonically increasing with the discriminatory severity of a rule.

3.1.2 DAAR

DAAR uses DCI together with minimum confidence and support to efficiently select non-discriminatory rules. DAAR's algorithm is shown in Figure 1.

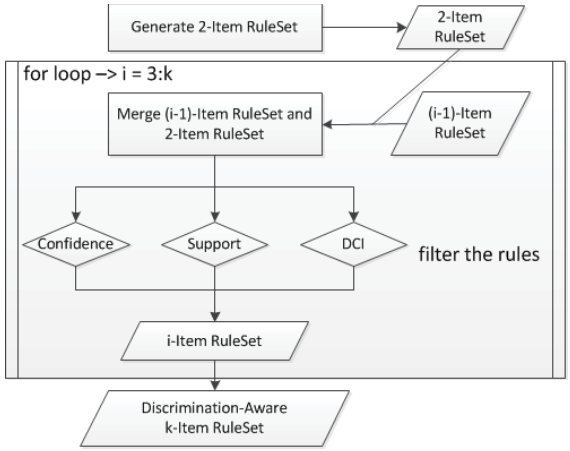


Figure 1. DAAR's Algorithm

DAAR starts from the set of 2-item rules (i.e. the rules with one attribute value and the class attribute), which is the base case, and merges with other 2-item rules iteratively until it gets the k-item rules, where k is the upper bound for the number of items in the rule. In each iteration, the rules are filtered by confidence, support and DCI. To classify new instances, DAAR uses majority voting based on the number of rules that predict the same class. If the vote is tied, the DCI sum for all rules for each class is compared and the class with lower sum (i.e. less discrimination) is selected.

3.2 Decision Tree and DADT

Decision Trees (DTs) are one of the most popular machine learning algorithms. The standard DT algorithm uses information gain to select the best attribute at each step as a root of the tree/subtree, until all examples in the subset belong to the same class, in which case it creates a leaf node labelled with this class. DTs can be seen as generating a set of mutually exclusive rules – each path from the root of the tree to a leaf node is one rule, and each rule is a conjunction of attribute tests. DADT is a discrimination-aware version of DT introduced by Faisal et al. in [3]. The tree is constructed in two phases. In the first phase, it generates a tree by using a new splitting criterion: IGC-IGS. IGC is the standard information gain (Information Gain regarding the

Class label) and IGS is Information Gain regarding the Sensitive attribute, defined as:

$$IGS = H_S(D) - \sum_{j=1}^{k} \frac{|D_j|}{|D|} H_S(D_j)$$

where S is the sensitive attribute, $H_S(D) = -\sum_{i=1}^{n} P_i * \log_2 P_i$ is the entropy of set D with respect to S and P_i is the proportion of items with the i^{th} value of the sensitive attribute.

As the aim is to have higher IGC but lower IGS, the difference IGC-IGS is an appropriate criterion. In the second phase, the leaves are relabeled to decrease the discrimination severity to less than ε (where ε is a non-discriminatory constraint), while sacrificing as little accuracy as possible. Experiments on census income datasets showed that DADT can produce a tree with a lower discrimination while maintaining accuracy [3].

4. EXPERIMENTS AND RESULTS

We consider the task of predicting exam performance in a fist year programming course. We compare the performance of the discrimination-aware classifiers DAAR and DADT with their standard non-discrimination-aware counterparts CBA (standard AR) and C4.5 (standard DT).

4.1 Dataset and Experimental Setup

Learning computer programming is difficult as it requires a lot of practice with feedback, and a very precise way of thinking. It is easy for students to fall behind, especially since introductory computer programming courses have a large number of students. Predicting students at risk of failing or not performing well is highly desirable.

Our evaluation is conducted using data from a first year computer programming course at an Australian University with 220 students. Our goal is to predict the exam performance, *high* or *low*, based on the student grades on the assessments during the semester and some enrolment attributes such as country of residence, degree name and if the student is local or international. A description of the attributes and their values is given in Table 1.

Table 1. Description of Attributes

Attribute	Description	Number of Attribute Values
Country	Country of permanent residence: {Australia, Brazil, China, …}	26
Degree	Name of the degree the student is enrolled into: {Bachelor of Science, Bachelor of Engineering,…}	27
Local	Indicates if the student is Australian or not: {Local, International}	2
a1_grade	The grade of assessment 1 during semester: {HD, D, CR, P, F}	5
a2_grade	The grade of assessment 2 during semester: {HD, D, CR, P, F}	5
a3_grade	The grade of assessment 3 during semester: {HD, D, CR, P, F}	5
a4_grade	The grade of assessment 4 during semester: {HD, D, CR, P, F}	5
a5_grade	The grade of assessment 5 during semester: {HD, D, CR, P, F}	5
Exam	Exam performance during examination period: {high, low}	2

The grades for the 5 assessments during the semester are the standard grades used at the university defined as follows: *HD* (High Distinction, mark of [85, 100]), *D* (Distinction, mark of [75, 84]), *CR* (Credit, mark of [65, 74]), *P* (Pass, mark of [50, 64]) and *F* (Fail, mark below 50). The exam performance is defined as *high* if the exam mark is 65 or higher (i.e. HD, D or CR), and *low* if it is below 65 (i.e. P or F). There were 105 students in the *high* group and 115 in the *low* group.

We selected the exam grade as a variable to predict rather than the final grade in the course, as the exam is the major assessment component (worth 50% and covering all topics) and it is also independent of the assessment components during the semester, while these components contribute to calculating the final grade for the course.

Among the 8 predictors, we consider *country* as the sensitive attribute, which means that we would like to avoid discrimination based on the student nationality. Originally, this attribute had 26 different values, with 5 or less number of students for most of the countries, so we aggregated these values into three groups: *Australia*, *China* and *Others*. The number of students in each group was 127, 54 and 39, respectively.

4.2 Results and Discussion

To evaluate the performance of the classification methods, we use 10-fold cross validation in all experiments. We report both the average value and the standard deviation for the 10 folds. As predictive accuracy measures, we use both classification accuracy and F-measure.

To assess the discrimination severity of the classifier, we calculate a discrimination score. In [1] a discrimination score for a binary sensitive attribute S with values S_1 and S_2, and class values C_+ and C_- is defined as:

$$\text{Score} = |\mathrm{P}(\mathrm{C} = \mathrm{C}_+ | \mathrm{S} = \mathrm{S}_1) - \mathrm{P}(\mathrm{C} = \mathrm{C}_+ | \mathrm{S} = \mathrm{S}_2)|$$

As our sensitive attribute has three values, we extend this definition to multi-valued attribute with m ($m>2$) values. We compute the score for each value S_i and then average the m scores:

$$\text{Score} = \frac{1}{\mathrm{m}} * \left(\sum_{\mathrm{i}=1}^{\mathrm{m}} |\mathrm{P}(\mathrm{C} = \mathrm{C}_+ | \mathrm{S} = \mathrm{S}_\mathrm{i}) - \mathrm{P}(\mathrm{C} = \mathrm{C}_+ | \ \mathrm{S} = \mathrm{S}_{\mathrm{others}})| \right)$$

where S_{others} represents all the attribute values other than S_i.

If the score is 0, there is no discrimination. Otherwise, a higher score corresponds to a higher discrimination severity.

4.2.1 DAAR

Table 2 presents the accuracy results and discrimination score for the standard AR and DAAR. We can see that DAAR was able to decrease the discrimination score of AR from 0.2831 to 0.2653. The trade-off was a slightly lower accuracy - DAAR achieved 73.92% accuracy, which is 4.72% lower than AR's accuracy.

Table 2. Results for Standard AR and DAAR

	Standard AR		DAAR	
	Mean	**Std.**	**Mean**	**Std.**
Accuracy	78.64%	0.0037	73.92%	0.0128
F-measure	0.7863	0.0037	0.7389	0.0131
Disc. score	0.2831	0.0109	0.2653	0.0163

Table 3 shows some representative and interesting rules produced by DAAR with their confidence, support and DCI. These rules are very compact, easy to understand and apply by teachers.

Table 4 shows the rules with high confidence and support that were filtered out by DAAR, as they were discriminatory with respect to the sensitive attribute *country*.

Table 3. Sample Rules Produced by DAAR

Rules	Conf.	Sup.	DCI
a1_grade=CR → exam = low	1.0	0.01	0
degree= Bachelor of Commerce, a4_grade=HD → exam = high	1.0	0.01	0
a4_grade=F → exam = low	1.0	0.19	0
degree= Bachelor of Engineering & Bachelor of Science, a5_grade=HD → exam = high	0.84	0.08	0

Table 4. Discriminatory Rules Removed by DAAR

Rules	Conf.	Sup.	DCI
country=Other → exam = high	0.62	0.12	0.17
country=CH → exam = low	0.77	0.18	0.26
country=Others, a5_grade=HD, a4_grade=HD → exam = high	0.83	0.08	0.17

4.2.2 DADT

The trees produced by the standard DT and DADT are shown in Figure 2 and Figure 3, respectively. The standard DT achieved accuracy of 83.46% but it used the sensitive attribute *country* and its discrimination score was 0.2298. DADT achieved a slightly lower accuracy of 82.73% without using the sensitive attribute. Thus, DADT is able to avoid discrimination with a minimum loss in accuracy. Both DTs included the attribute *a4_grade* as a root of the tree, which shows the importance of this attribute for predicting exam performance.

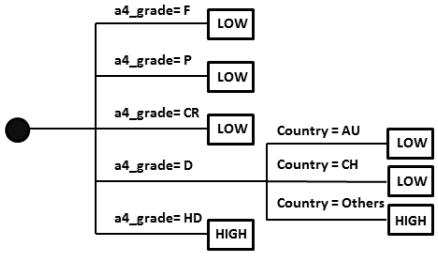


Figure 2. Tree Produced by the Standard DT

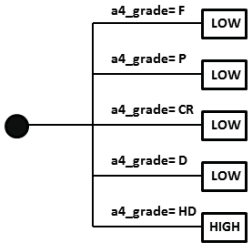


Figure 3. Tree Produced by DADT

4.2.3 Discussion

In terms of overall performance, all four methods had reasonable accuracy, from 73.92% to 83.46%, with the DT-based classifiers outperforming the AR-based classifiers. All classifiers generated a small set of rules that are easy to understand and use by teachers. The AR classifiers used more attributes in the rules which, for our case study, provided additional insights about the important attributes in predicting student performance and providing feedback to students.

In terms of discrimination, we can see that both DAAR and DADT decreased the severity of the discrimination compared to their standard counterparts, with trivial loss in accuracy. Specifically, DAAR removed the rules with higher DCI values and reduced the discrimination score, and DADT using IGC-IGS as an attribute selection criterion, built a DT without using the sensitive attribute *country*.

5. CONCLUSIONS

Educational data often contains sensitive attributes, which should only be used as information carriers rather than factors to distinguish students and potentially discriminate them. We investigated discrimination-aware classification for mining of educational data, with a case study in predicting student exam performance based on enrolment information and assessment marks during the semester, in the context of a computer programming course. We applied our discrimination-aware method DAAR, which is based on association rules, and also DADT, a discrimination-aware decision tree method, and compared DAAR and DADT with their non-discrimination-aware alternatives. The experiment results showed that both DAAR and DADT decreased the discimination with minor impact on the predictive accuracy. Both classifiers generated a small set of rules that are easy to understand and use by teachers and students. The discrimination-aware classifiers can be used for any classification tasks in educational settings, such as identifying students at risk, to provide timely feedback and intervention.

6. REFERENCES

[1] Calders, T. and Verwer, S., 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery 21*, 2, 277-292.

[2] Pedreshi, D., Ruggieri, S., and Turini, F., 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* ACM, 560-568.

[3] Kamiran, F., Calders, T., and Pechenizkiy, M., 2010. Discrimination aware decision tree learning. In *Proceedings of the 10th IEEE International Conference on Data Mining* IEEE, 869-874.

[4] Luo, L., Liu, W., Koprinska, I., and Chen, F., 2015. Discrimination-Aware Association Rule Mining for Unbiased Data Analytics. *TR700*, School of Information Technologies, The University of Sydney.

[5] Romero, C., Ventura, S., Espejo, P.G., and Hervás, C., 2008. Data Mining Algorithms to Classify Students. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 8-17.

[6] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S., 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education 68*, 458-472.

[7] Kotsiantis, S., Patriarcheas, K., and Xenos, M., 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems 23*, 6, 529-535.

[8] Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V., 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education 53*, 3, 950-965.

[9] Kamiran, F. and Calders, T., 2009. Classifying without discriminating. In *International Conference on Computer, Control and Communication* IEEE, 1-6.

[10] Ma, Y., Liu, B., and Yiming, W.H., 1998. Integrating classification and association rule mining. In *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 80-86.

Language to Completion: Success in an Educational Data Mining Massive Open Online Class

Scott Crossley
Georgia State U.
Atlanta, GA 30303
scrossley@gsu.edu

Danielle S. McNamara
Arizona State Univ.
Tempe, AZ, 85287
dsmcnama@asu.edu

Ryan Baker, Yuan Wang, Luc Paquette
Teachers College
Columbia University
New York, NY 10027
ryanshaunbaker@gmail.com

Tiffany Barnes
NC State Univ.
Raleigh, NC 27606
tmbarnes@ncsu.edu

Yoav Bergner
Educational Testing Service
Princeton, NJ 08541
ybergner@ets.org

ABSTRACT

Completion rates for massive open online classes (MOOCs) are notoriously low, but learner intent is an important factor. By studying students who drop out despite their intent to complete the MOOC, it may be possible to develop interventions to improve retention and learning outcomes. Previous research into predicting MOOC completion has focused on click-streams, demographics, and sentiment analysis. This study uses natural language processing (NLP) to examine if the language in the discussion forum of an educational data mining MOOC is predictive of successful class completion. The analysis is applied to a subsample of 320 students who completed at least one graded assignment and produced at least 50 words in discussion forums. The findings indicate that the language produced by students can predict with substantial accuracy (67.8 %) whether students complete the MOOC. This predictive power suggests that NLP can help us both to understand student retention in MOOCs and to develop automated signals of student success.

Keywords

Natural language processing, MOOCs, student success

1. INTRODUCTION

The sheer size of student populations in massive open online classes (MOOCs) requires educators to rethink traditional approaches to instructor intervention and the assessment of student motivation, engagement, and success [11]. As a result, a good deal of MOOC research has focused on predicting or explaining attrition and overall student success. Most research assessing student success in MOOCs has involved the examination of click-stream data. Such data provides researchers with evidence of engagement within the course and activities associated with individual course goals [6]. Additional approaches to assessing student success include the use of sentiment analysis tools to gauge students' affective states [15, 16] and individual difference measures such as student backgrounds and other demographic variables [5].

In this paper, we explore the potential for natural language processing (NLP) tools that include but also go beyond sentiment analysis to predict success in an educational data mining MOOC. Our goal is to develop an automated model of MOOC success based on NLP variables such as text length, text cohesion, syntactic complexity, lexical sophistication, and writing quality that can be used to predict class completion. Thus, in line with Koller et al. [7], we hope to better understand the language produced by MOOC students, especially differences in the language between those students that complete a course and those that do not. Using NLP variables affords the opportunity to go beyond click-stream data to examine student success and allows the personalization of predictive variables based solely on the language differences exhibited by students. Such fine-grained content analyses may allow teachers to monitor and detect evidence of student engagement, emotional states, and linguistic ability to predict success and intervene to prevent attrition.

1.1 NLP and MOOC Success

Researchers and teachers have embraced MOOCs for their potential to increase accessibility to distance and lifelong learners [7]. From a research perspective, MOOCs provide a tremendous amount of data via click-stream logs within the MOOC platform. These data can be mined to investigate student learning, student completion, and student attitudes. Typical measures include frequency of access to various learning resources, time-on-task, or attempt rates on graded assignments [14]. Less frequently mined, however, are data related to language use [15, 16].

NLP refers to the examination of texts' linguistic properties using a computational approach. NLP centers on how computers can be used to understand and manipulate natural language texts (e.g., student posts in a MOOC discussion forum) to do useful things (e.g., predict success in a MOOC). The principal aim of NLP is to gather information about human language understanding and production through the development of computer programs intended to process and understand language in a manner similar to humans [3]. Traditional NLP tools focus on a text's syntactic and lexical properties, usually by counting the length of sentences or words or using databases to compare the contents of a single text to that of a larger, more representative corpus of texts. More advanced tools provide measurements of text cohesion, the use of rhetorical devices, syntactic similarity, and more sophisticated indices of word use.

In MOOCs, the most common NLP approach to analyzing student language production has been through the use of sentiment analysis tools. Such tools examine language for positive or negative emotion words or words related to motivation, agreement, cognitive mechanisms, or engagement. For instance, Wen et al. [16] examined the sentiment of forum posts in a MOOC to examine trends in students' opinions toward the course and course tools. Using four variables related to text sentiment (words related to application, cognitive words, first person pronouns, and positive words), Wen et al. reported that students' use of words related to motivation had a lower risk of dropping out of the course. In addition, the more students used personal pronouns in forum posts, the less likely they were to drop out of the course. In a similar study, Wen et al [15] reported a significant correlation between sentiment variables and the number of students who dropped from a MOOC on a daily basis. However, Wen et al. did not report a consistent relation between students' sentiment across individual courses and dropout rates (e.g., in some courses negative words such as "challenging" or "frustrating" were a sign of engagement), indicating a need for caution in the interpretation of sentiment analysis tools.

2. METHOD

The goal of this study is to examine the potential for NLP tools to predict success in an EDM MOOC. Specifically, we examine the language used by MOOC students in discussion forums and use this language to predict student completion rates.

2.1 The MOOC: Big Data in Education

The MOOC of interest for this study is the Big Data in Education MOOC hosted on the Coursera platform as one of the inaugural courses offered by Columbia University. It was created in response to the increasing interest in the learning sciences and educational technology communities in learning to use EDM methods with fine-grained log data. The overall goal of this course was to enable students to apply each method to answer education research questions and to drive intervention and improvement in educational software and systems. The course covered roughly the same material as a graduate-level course, Core Methods in Educational Data Mining, at Teachers College Columbia University. The MOOC spanned from October 24, 2013 to December 26, 2013. The weekly course comprised lecture videos and 8 weekly assignments. Most of the videos contained in-video quizzes (that did not count toward the final grade).

All the weekly assignments were automatically graded, numeric input or multiple-choice questions. In each assignment, students were asked to conduct an analysis on a data set provided to them and answer questions about it. In order to receive a grade, students had to complete this assignment within two weeks of its release with up to three attempts for each assignment, and the best score out of the three attempts was counted. The course had a total enrollment of over 48,000, but a much smaller number actively participated; 13,314 students watched at least one video; 1,242 students watched all the videos; 1,380 students completed at least one assignment; and 710 made a post in the weekly discussion sections. Of those with posts, 426 completed at least one class assignment; 638 students completed the online course and received a certificate (meaning that some students could earn a certificate without participating in the discussion forums at all).

2.2 Student Completion Rates

We selected completion rate as our variable of success because it is one of the most common metrics used in MOOC research [17]. However, as pointed out by several researchers, learner intent is a critical issue [5, 6, 7]. Many MOOC students enroll based on curiosity, with no intention of completing the course. The increased use of entry surveys is no doubt related to this inference problem. In the present analysis, however, we do not have access to this information. Therefore, we compute completion rates based on a smaller sample of forum posters as described below. "Completion" was pre-defined as earning an overall grade average of 70% or above. The overall grade was calculated by averaging the 6 highest grades extracted out of the total of 8 assignments.

2.3 Discussion Posts

We selected discussion posts because they are one of the few instances in MOOCs that provide students with the opportunity to engage in social learning [11, 16]. Discussion forums provide students with a platform to exchange ideas, discuss lectures, ask questions about the course, and seek technical help, all of which lead to the production of language in a natural setting. Such natural language can provide researchers with a window into individual student motivation, linguistics skills, writing strategies, and affective states. This information can in turn be used to develop models to improve student learning experiences [11]. In the EDM MOOC, students and teaching staff participated in weekly forum discussions. Each week, new discussion threads were created for each week's content including both videos and assignments under sub-forums. Forum participation did not count toward student's final grades. For this study, we focused on the forum participation in the weekly course discussions.

For the 426 students who both made a forum post and completed an assignment, we aggregated each of their posts such that each post became a paragraph in a text file. We selected only those students that produced at least 50 words in their aggregated posts (n = 320). We selected a cut off of 50 words in order to have sufficient linguistic information to reliably assess the student's language using NLP tools. Of these 320 students, 132 did not successfully complete the course while the remaining 188 students completed the course.

2.4 Natural Language Processing Tools

We used several NLP tools to assess the linguistic features in the aggregated posts of sufficient length. These included the Writing Assessment Tool (WAT [9]), the Tool for the Automatic Analysis of Lexical Sophistication (TAALES [8]), and the Tool for the Automatic Assessment of Sentiment (TAAS). We provide a brief description of the indices reported by these tools below.

2.4.1 WAT

WAT was developed specifically to assess writing quality. As such, it includes a number of writing specific indices related to text structure (text length, sentence length, paragraph length), cohesion (e.g., local, global, and situational cohesion), lexical sophistication (e.g., word frequency, age of acquisition, word hypernymy, word meaningfulness), key word use, part of speech tags (adjectives, adverbs, cardinal numbers), syntactic complexity, and rhetorical features. It also reports on a number of writing quality algorithms such as introduction, body, and conclusion paragraph quality and the overall quality of an essay.

2.4.2 TAALES

TAALES incorporates about 150 indices related to basic lexical information (e.g., the number of tokens and types), lexical frequency, lexical range, psycholinguistic word information (e.g., concreteness, meaningfulness), and academic language for both single words and multi-word units (e.g., bigrams and trigrams).

2.4.3 TAAS

TAAS was developed specifically for this study. The tool incorporates a number of language-based sentiment analysis databases including the Linguistic Inquiry and Word Count database (LIWC [10]), Affective Norms for English Words (ANEW [1]), Geneva Affect Label Coder (GALC [13]), the National Research Council (NRC) Word-Emotion Association Lexicon [12], and the Senticnet database [2]. Using these databases, TAAS computes affective variables related to a number of emotions such as anger, amusement, fear, sadness, surprise, trust, pleasantness, attention, and sensitivity.

2.5 Statistical Analysis

The indices reported by WAT, TAALES, and TAAS that yielded non-normal distributions were removed. A multivariate analysis of variance (MANOVA) was conducted to examine which indices reported differences between the postings written by students who successfully completed the course and those who did not. The MANOVA was followed by stepwise discriminant function analysis (DFA) using the selected NLP indices that demonstrated significant differences between those students who completed the course and those who did not, and did not exhibit multicollinearity ($r > .90$) with other indices in the set. In the case of multicollinearity, the index demonstrating the largest effect size was retained in the analysis. The DFA was used to develop an algorithm to predict group membership through a discriminant function co-efficient. A DFA model was first developed for the entire corpus of postings. This model was then used to predict group membership of the postings using leave-one-out-cross-validation (LOOCV) in order to ensure that the model was stable across the dataset.

3. RESULTS

3.1 MANOVA

A MANOVA was conducted using the NLP indices calculated by WAT, TAALES, and TAAS as the dependent variables and the postings by students who completed the course and those who did not as the independent variables. A number of indices related to positing length, number of posts, use of numbers, writing quality, lexical sophistication, n-gram use, and cohesion demonstrated significant differences (see Table 1 for the MANOVA results). These indices were used in the subsequent DFA.

The results indicate that those who completed the course, even though course completion depended solely on success on technical assignments, tended to be better writers (i.e., received higher scores based on the essay score algorithm in WAT), to use a greater variety of words, to write more often with more words, and with greater cohesion. They also used more words relevant to the domain of the course, more concrete words, more sophisticated words, words with more associations to other words, and more common bigrams and trigrams.

3.2 Discriminant Function Analysis

A stepwise DFA using the indices selected through the MANOVA retained seven variables related to post length, lexical sophistication, the use of numbers, cohesion, and writing quality as significant predictors of whether a student received a certificate or not. These indices were *Average post lengths, Word age of acquisition, Cardinal numbers, Hypernymy standard deviation, Situational cohesion, Trigram frequency,* and *Essay score algorithm*. The remaining variables were removed as non-significant predictors.

Table 1. MANOVA Results Predicting Whether Students Completed the MOOC

Index	*F*	*η2*
Essay score algorithm	13.071**	0.039
Type token ratio	12.074**	0.037
Number of word types	11.371**	0.035
Number of posts	10.919*	0.033
Average post length	10.596*	0.032
Concreteness	10.017*	0.031
Cardinal numbers	10.081*	0.031
Trigram frequency	9.445*	0.029
Bigram frequency	8.903*	0.027
Number of sentences	8.451*	0.026
Frequency content words	8.219*	0.025
Situational cohesion	8.041*	0.025
Hypernymy standard deviation	7.643*	0.023
Word meaningfulness	7.378*	0.023
Lexical diversity	6.180*	0.019
Average word length	5.150*	0.016
Essay body quality algorithm	4.409*	0.014
Logical connectors	3.915*	0.012
Word age of acquisition	3.854*	0.012

** $p < .001$, * $p < .050$

The results demonstrate that the DFA using these seven indices correctly allocated 222 of the 320 posts in the total set, $\chi 2$ (df=1) = 46.529 $p < .001$, for an accuracy of 69.4%. For the leave-one-out cross-validation (LOOCV), the discriminant analysis allocated 217 of the 320 texts for an accuracy of 67.8% (see the confusion matrix reported in Table 2 for results and F_1 scores). The Cohen's Kappa measure of agreement between the predicted and actual class label was 0.379, demonstrating fair agreement.

Table 2. Confusion matrix for DFA classifying postings

		predicted		
	actual	- Cert	+Cert	F_1 score
Whole set	- Certificate	**91**	41	0.650
	+Certificate	57	**131**	0.728
LOOCV	- Certificate	**87**	45	0.628
	+Certificate	58	**130**	0.716

4. DISCUSSION AND CONCLUSION

Previous MOOC studies have investigated completion rates though click-stream data and sentiment analysis tools. The current study adds another tool for examining successful completion of a MOOC: natural language processing. The tools assessed in this study show that language related to forum post length, lexical sophistication, situational cohesion, cardinal numbers, trigram production, and writing quality can significantly predict whether a MOOC student completed an EDM course. Such a finding has important implications for how students' individual differences (in this case, language skills) that go beyond observed behaviors (i.e., click-stream data) can be used to predict success.

Overall, the results support the basic notion that students that demonstrate more advanced linguistic skills, produce more coherent text, and produce more content specific posts are more likely to complete the EDM MOOC. For instance, students were more likely to complete the course if their posts were shorter (i.e., more efficient), used words that are less frequent or familiar (i.e., higher age of acquisition scores), used more cardinal numbers (i.e., content specific), used words that were more consistent in

terms of specificity (i.e., less variance in terms of specificity), produced posts that were more cohesive (i.e., greater overlap of ideas), used more frequent trigrams (i.e., followed expected combinations of words), and produced writing samples of higher quality (i.e., samples scored as higher quality by a automatic essay scoring algorithm). Interestingly, none of our affective variables distinguished between students who completed or did not complete the EDM MOOC. This may be the result of the specific MOOC under investigation, a weakness of the affective variables examined, or a weakness of affective variables in general.

The findings have important practical implications as well. The linguistic model developed in this paper through the DFA could be used as a prototype to monitor MOOC students and potentially identify those students who are less likely to complete the course. Such students could then be target for interventions (e.g., sending e-mails, suggesting assignments or tutoring) to improve immediate engagement in the MOOC and promote long-term completion.

The results reported in this study are both significant and extendible to similar datasets (as reported in the LOOCV results). They also open up additional research avenues. For instance, to improve detection of students who might be unlikely to complete the MOOC, follow-up models that include click-stream data could be developed and tested. Such models would likely provide additive power to detection accuracy. One concern with the current model is that it requires language samples for analysis. This suggests that NLP approaches like this one may be even more useful in classes that have activities such as collaborative chat, a feature now emerging in some MOOCs.

5. ACKNOWLEDGMENTS

This research was supported in part by the Institute for Education Sciences and National Science Foundation (IES R305A080589, IES R305G20018-02, and DRL- 1418378). Ideas expressed in this material are those of the authors and do not necessarily reflect the views of the IES or the NSF.

6. REFERENCES

[1] Bradley, M. M., and Lang, P. J. 1999. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. *Technical report. The Center for Research in Psychophysiology, University of Florida*.

[2] Cambria, E. and Hussain, A. 2015. *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Cham, Switzerland: Springer.

[3] Crossley, S. A. 2013. Advancing research in second language writing through computational tools and machine learning techniques: A research agenda. *Language Teaching, 46* (2), 256-271.

[4] DeBoer, J., Stump, G. S., Seaton, D., Ho, A., Pritchard, D. E., and Breslow, L. 2013. Bringing student backgrounds online: MOOC user demographics, site usage, and online learning. *In the Proceedings of the 6th International Conference on Educational Data Mining*, 312-313.

[5] DeBoer, J., Ho, A. D., Stump, G. S., & Breslow, L. 2014. Changing "Course": Reconceptualizing Educational Variables for Massive Open Online Courses. *Educational Researcher*, *March*, 74–84.

[6] Kizilcec, R. F., Piech, C., and Schneider, E. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. *In the Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 170-179.

[7] Koller, D., Ng, A., Do, C., and Chen, Z. 2013. Retention and Intention in Massive Open Online Courses. *Educause*.

[8] Kyle, K., and Crossley, S. A. in press. Automatically Assessing Lexical Sophistication: Indices, Tools, Findings, and Application. *TESOL Quarterly.*

[9] McNamara, D. S., Crossley, S. A., & Roscoe, R. 2013. Natural Language Processing in an Intelligent Writing Strategy Tutoring System. *Behavior Research Methods, 45* (2), 499-515.

[10] Pennebaker, J. W., Booth, R. J., and Francis, M. E. 2007. *LIWC2007: Linguistic inquiry and word count.* Austin, Texas.

[11] Ramesh, A., Goldwasser, D., Huang, B., Daume, H., and Getoor, L. 2014. Understanding MOOC Discussion Forums using Seeded LDA. *ACL Workshop on Innovative Use of NLP for Building Educational Applications,* 22-27.

[12] Saif, M., and Turney, P. 2013. Crowdsourcing a Word-Emotion Association Lexicon**,** *Computational Intelligence,* 29 (3), 436-465.

[13] Scherer, K. R. 2005. What are emotions? And how should they be measured? *Social Science Information*, 44 (4), 695-729.

[14] Seaton, D. T., Bergner, Y., Chuang, I., Mitros, P., & Pritchard, D. E. (2014). Who does what in a massive open online course? *Communications of the ACM*, *57*(4), 58–65.

[15] Wen, M., Yang, D. and Rose, C. P. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? *In the Proceedings of the 7th International Conference on Educational Data Mining, 130-137*.

[16] Wen, M., Yang, D. and Rose, C. P. 2014. Linguistic Reflections of Student Engagement in Massive Open Online Courses. *In the Proceedings of the International Conference on Weblogs and Social Media*.

[17] Wang, Y. 2014. MOOC Leaner Motivation and Learning Pattern Discovery. *In the Proceedings of the 7th International Conference on Educational Data Mining*, 452-454.

A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance

Pedro Strecht
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
pstrecht@fe.up.pt

Luís Cruz
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
luiscruz@fe.up.pt

Carlos Soares
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
csoares@fe.up.pt

João Mendes-Moreira
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
jmoreira@fe.up.pt

Rui Abreu
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
rma@fe.up.pt

ABSTRACT

Predicting the success or failure of a student in a course or program is a problem that has recently been addressed using data mining techniques. In this paper we evaluate some of the most popular classification and regression algorithms on this problem. We address two problems: prediction of approval/failure and prediction of grade. The former is tackled as a classification task while the latter as a regression task. Separate models are trained for each course. The experiments were carried out using administrate data from the University of Porto, concerning approximately 700 courses. The algorithms with best results overall in classification were decision trees and SVM while in regression they were SVM, Random Forest, and AdaBoost.R2. However, in the classification setting, the algorithms are finding useful patterns, while, in regression, the models obtained are not able to beat a simple baseline.

Keywords

Regression, Classification, Academic Performance

1. INTRODUCTION

Recently, the University of Porto (UPorto) identified modelling of the success/failure of students in each course as one of its priorities. The goal is to use the models for two tasks: make predictions for the individual performance of students in courses and understand the factors associated with success and failure. These models are relevant to five levels of decision, namely: Course teacher, Program Director, Department Director, Faculty Director and University Rector. Course teachers and program directors can use the models to identify students at risk and devise strategies that can reduce the risk of failure. Also, program directors as well as department directors can find them useful in designing program syllabus. Finally, the top levels of university management can use these models to understand general trends and behaviours in student performance, which can lead to new or adapted pedagogical strategies.

The fact that models are needed for different levels of decision requires that these models have different granularities. In other words, course teachers and program directors are able to work with a few or a few dozen models, respectively. However, the other levels of management would have to deal with hundreds, maybe even thousands of models, which is not feasible. On the other hand, each course presents different particularities which makes the creation of a unique model to predict academic success for all the courses, an extremely hard task. Such a model would have to aggregate the different factors that influence success in very different courses. Therefore, we train a model separately for each course.

So far, the results obtained and the domain-specific constraints provide a satisfactory justification for the choice of decision trees. However, there is a need to understand the impact of this choice in the predictive accuracy of the algorithms, namely when compared with others. Additionally, although the problem of predicting if a student will pass or fail (classification task) is relevant for all levels of management of the university, the related problem of predicting the actual grade (regression task) may provide additional useful information. Therefore, this study also considers a comparative analysis of different regression algorithms. This comparison will also address the question of whether the features that are useful for classification are equally useful for regression.

The main contributions of this paper are: 1) to compare the predictive accuracy of different algorithms on the problems of predicting the performance of students in both classification (predicting success/failure) and regression (predicting the grade) tasks, particularly when comparing with decision trees, which have some other properties that deem

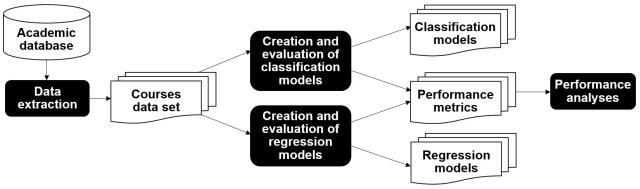


Figure 1: Experimental Setup

them suitable for this problem; 2) to assess whether the features which have obtained positive results in the classification task, and that represent essentially administrative information, are also useful to predict the grades.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 describes the experimental set-up and methodology for both classification and regression models. Section 4 presents the results followed by section 5 with the conclusions and future work.

2. RELATED WORK

Predicting students' performance has been an issue studied previously in educational data mining research in the context of student attrition [24, 23]. Minaei-Bidgoli [13] used a combination of multiple classifiers to predict their final grade based on features extracted from logged data in an education webbased system.

Pittman [15] performed a study to explore the effectiveness of data mining methods to identify students who are at risk of leaving a particular institution. Romero et al. [16] focused on comparing different data mining methods and techniques for classifying students based on their Moodle (e-learning system) usage data and the final marks obtained in their respective programmes. The conclusion was that the most appropriate algorithm was decision trees for being accurate and comprehensible for instructors. Kabakchieva [10] also developed models for predicting student performance, based on their personal, pre-university and university performance characteristics. The highest accuracy is achieved with the neural network model, followed by the decision tree model and the kNN model.

Strecht, Mendes-Moreira and Soares [20] work predicted the failure of students in university courses using an approach to group and merge interpretable models in order to replace them with more general ones. The results show that merging models grouped by scientific areas yields an improvement in prediction quality.

3. METHODOLOGY

To carry out the experiments, a system with four processes was developed following the architecture presented in Figure 1. The first process creates the data sets (one for each course in the university) from the academic database, containing enrolment data. The courses data set were then used by two processes to create classification and regression models for each course using various algorithms. These models were evaluated using suitable performance metrics (different for classification and regression) that are collected to allow analyses and comparison by the final process.

3.1 Data Extraction

This process extracts data sets from the academic database of the university information system. The analysis done focuses on the academic year 2012/2013. A total of 5779 course data sets were extracted (from 391 programmes). The variables used were: age, sex, marital status, nationality, displaced (whether the student lived outside the Porto district), scholarship, special needs, type of admission, type of student (regular, mobility, extraordinary), status of student (ordinary, employed, athlete, ...), years of enrolment, delayed courses, type of dedication (full-time, part-time), and debt situation. The target variables are approval for classification and final grade for regression.

The final grade in these data sets is stored as a numerical value between 0 and 20. However, there are some special cases in which the grade is given as an acronym (e.g, RA means fail because of dropout), which is not feasible for regression. In such cases, in which a student failed, we converted the grade to 0.

3.2 Creation and evaluation of models

Two processes trained a set of models for classification and regression respectively for each course using different algorithms. For classification we have used k-Nearest Neighbors (kNN) [9], Random Forest (RF) [2], AdaBoost (AB) [7], Classification and Regression Trees (CART) [3], Support Vector Machines [21], Naïve Bayes (NB) [12] and for regression we used Ordinary Least Squares (OLS) [18], SVM, CART, kNN, Random Forest, and AdaBoost.R2 (AB.R2) [8].

This selection of algorithms was based on the most used algorithms for general data mining problems [22]. In this set of experiments a standard values of parameters was used. As baseline in classification we defined a model which always predicts failure. For regression, the baseline model predicts the average grade of the training set of a given course.

Models were evaluated using the k-fold cross-validation method [19] with stratified sampling [11]. The distribution of positive and negative instances is not balanced, thus it is necessary to ensure that the distribution of students in each fold respect these proportions. Failure is the positive class in this problem and we used F1 score for evaluation [5]. All regression models used 10-fold cross validation and the Root Mean Squared Error (RMSE) as evaluation measure [4].

Training and evaluation of models was replicated for each course. Courses with less then 100 students were skipped. This resulted in around 700 models for each algorithm in both classification and regression.

3.3 Performance Analyses

In both classification and regression, the algorithms were compared by placing box plots side by side relating to F1 and RMSE respectively. To get a better perspective of the distribution of results, violin plots are presented together with the box plots. The longest horizontal lines inside the boxes refer to the median while the shortest refer to the average. A few descriptive statistics were also collected and presented in tables.

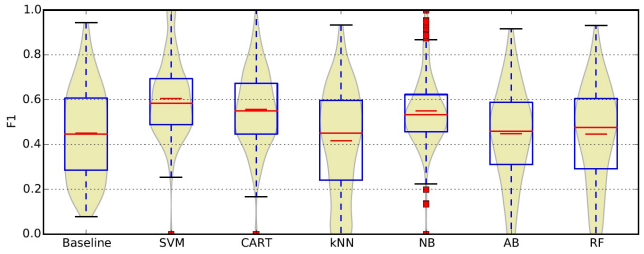


Figure 2: F1 score for each classification algorithm

In order to statistically validate the results obtained in the experiments we have used the Friedman test as suggested by Demšar to compare multiple classifiers [6]. We have used the typical value of 12 groups of models often referred as data sets in this context.

4. RESULTS

This section presents the results obtained by running experiments to train models for both classification and regression.

4.1 Classification

Figure 2 presents the F1 score distribution of models across algorithms. Table 1 presents some basic statistics about the results. Algorithms are ranked by descending order of values of the average and standard deviation of F1 scores.

The first fact that stands out from Figure 2 is that none of the algorithms present exceptional results. Albeit this, some of them seem to systematically outperform the baseline, namely SVM, CART and NB.

Table 1 confirms that SVM is the algorithm with the best performance, clearly outperforming the baseline. Not only it provides the highest average F1 score, 0.60±0.17, but sometimes it also achieves a maximum F1 score of 1.0, while the maximum score of the baseline is 0.94. Finally, although the minimum score is lower than the baseline's (0 vs. 0.08), the standard deviation is lower (0.17 vs. 0.20) which indicates that overall, it obtains more robust results.

Similar observations can be made for CART and NB. The performance of RF and AB is very similar to that of the baseline, while kNN is worse. The results of Random Forest, in particular, are surprising as this algorithm usually exhibits a very competitive performance [17].

In spite of the showing some systematic differences, the results are, overall, not very different. This is confirmed by the results of the Friedman test, $\chi^2(6) = 2.6071, p = 0.8563$, as the p-value is very high.

4.2 Regression

Figure 3 presents the distribution of RMSE values of models obtained by the algorithms. Table 2 presents some basic statistics about the results. The algorithms are ranked by ascending order of RMSE values.

As in classification, it is also quite straightforward that none of the algorithms present exceptional results. Also in this case, there is one algorithm which performs clearly worse

Table 1: Classification models results (F1)

Rank	Algorithm	Avg	Std Dev	Max	Min
1	SVM	0.60	0.17	1.00	0.00
2	CART	0.56	0.17	1.00	0.00
3	NB	0.55	0.16	1.00	0.00
4	RF	0.45	0.22	0.93	0.00
5	AB	0.45	0.21	0.92	0.00
6	Baseline	0.45	0.20	0.94	0.08
7	kNN	0.42	0.24	0.93	0.00

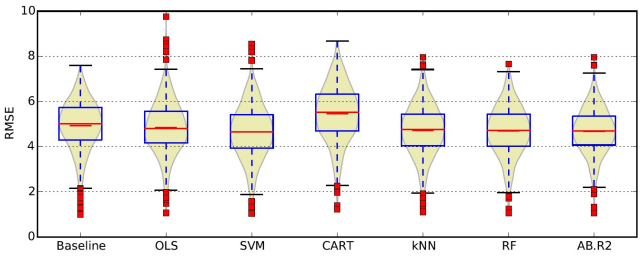


Figure 3: RMSE for each regression algorithm

than the baseline, CART (Table 2). Unlike classification, all violin plots show exactly the same shape, i.e., equally sized upper and lower tails. Therefore, differences are more related to overall performance (i.e. location). This shows that to compare models it is enough to consider the average and standard deviation.

The differences in performance are even smaller than in classification. However, Table 2 suggests that SVM was the best algorithm with an average of 4.65±1.19, but the standard deviation is quite large (1.19) taking into account the RMSE of the baseline (4.92). These observations are confirmed by the Friedman test ($\chi^2(6) = 3.3697, p = 0.7612$). In the case of regression, the value of the RMSE is interpretable, as it is in the same scale as the target variable. All algorithms obtain an error around 5, which is very high according to the scale (0 to 20).

In light of the results obtained in the classification setting, this is somewhat surprising, since the independent variables are the same and many of the algorithms used are based on the same principles.[1] Further analysis of the results is necessary to understand them and to identify possibilities to improve the results.

[1] Although this must be interpreted carefully as it is arguable to say that, for instance, SVM for classification and regression are the same algorithm.

Table 2: Regression models results (RMSE)

Rank	Algorithm	Avg	Std Dev	Max	Min
1	SVM	4.65	1.19	8.54	1.03
2	RF	4.69	1.10	7.66	1.06
3	AB.R2	4.69	1.02	7.96	1.07
4	kNN	4.72	1.12	7.96	1.10
5	Baseline	4.92	1.11	7.59	1.00
6	OLS	4.84	1.19	9.75	1.06
7	CART	5.46	1.26	8.68	1.22

5. CONCLUSIONS

Positive results were obtained on the classification approach where the goal is to predict whether a student will pass or fail a course. Surprisingly, however, the results on the regression approach, where the goal is to predict the grade of the student in a course, were bad. Additionally, we found no statistical evidence that the differences in performance between the algorithms are significant, although some trends are observed. Further analysis is necessary to better understand these results, which could lead to ideas for improvement. As a complement of the problems studied in this work, it should be interesting to predict an interval for a grade [1].

Some algorithms are more sensitive to parameter tuning than others. Thus it is not guaranteed that they ran with the best configuration. As future work, some optimisation could be made using an automate tuning methodology. In addition, feature selection and feature weighting can be carried out which has proven to yield good results in educational data [14].

Although the feature set used in the experiments provided some interesting results in classification, the same did not happen in regression. Thus, new features could be added. Features related to academic goals, personal interests, time management skills, sports activities, sleep habits, etc. are worthwhile investigating.

6. ACKNOWLEDGMENTS

This work is funded by projects "NORTE-07-0124-FEDER-000059" and "NORTE-07-0124-FEDER-000057", financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

7. REFERENCES

[1] R. Asif, A. Merceron, and M. Pathan. Predicting student academic performance at degree level: A case study. 2015.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[4] T. Chai and R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? *Geoscientific Model Development Discussions*, 7:1525–1534, 2014.

[5] N. Chinchor. MUC-4 Evaluation Metrics. In *Proceedings of the 4th Message Understanding Conference (MUC4 '92)*, pages 22–29. Association for Computational Linguistics, 1992.

[6] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[7] T. G. Dietterich. Machine-learning research: four current directions. *AI magazine*, 18(4):97, 1997.

[8] H. Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.

[9] E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

[10] D. Kabakchieva. Predicting Student Performance by Using Data Mining Methods for Classification. *Cybernetics and Information Technologies*, 13(1):61–72, 2013.

[11] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Conference on AI (IJCAI)*, pages 1137–1145, San Mateo, CA, 1995. Morgan Kaufmann.

[12] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93, 1994.

[13] B. Minaei-Bidgoli. Predicting student performance: an application of data mining methods with an educational web-based system. In *33rd ASEE/IEEE Frontiers in Education Conference*, pages 1–6, 2003.

[14] B. Minaei-Bidgoli and W. F. Punch. Using genetic algorithms for data mining optimization in an educational web-based system. In *Genetic and Evolutionary Computation—GECCO 2003*, pages 2252–2263. Springer, 2003.

[15] K. Pittman. *Comparison of data mining techniques used to predict student retention*. PhD thesis, Nova Southeastern University, 2008.

[16] C. Romero. Data mining algorithms to classify students. In *1st International Educational Data Mining Conference (EDM08)*, 2008.

[17] M. R. Segal. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 2004.

[18] S. M. Stigler. Gauss and the invention of least squares. *The Annals of Statistics*, pages 465–474, 1981.

[19] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B*, 36(2):111–147, 1974.

[20] P. Strecht, J. Mendes-Moreira, and C. Soares. Merging Decision Trees: A Case Study in Predicting Student Performance. In X. Luo, J. Yu, and Z. Li, editors, *Advanced Data Mining and Applications*, Lecture Notes in Computer Science, pages 535–548. Springer International Publishing, 2014.

[21] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

[22] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[23] A. Zafra and S. Ventura. Predicting student grades in learning management systems with multiple instance genetic programming. *International Working Group on Educational Data Mining*, 2009.

[24] J. Zimmermann, K. H. Brodersen, J.-P. Pellet, E. August, and J. M. Buhmann. Predicting graduate-level performance from undergraduate achievements. In *4th International Educational Data Mining Conference (EDM11)*, pages 357–358, 2011.

Predicting Student Grade based on Free-style Comments using Word2Vec and ANN by Considering Prediction Results Obtained in Consecutive Lessons

Jingyi Luo
Graduate School of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

Shaymaa E.Sorour
Graduate School of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

Kazumasa Goda
Kyushu Institute of Information Science
Dazaifu,Japan

Tsunenori Mine
Faculty of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

ABSTRACT

Continuously tracking students during a whole semester plays a vital role to enable a teacher to grasp their learning situation, attitude and motivation. It also helps to give correct assessment and useful feedback to them. To this end, we ask students to write their comments just after each lesson, because student comments reflect their learning attitude towards the lesson, understanding of course contents, and difficulties of learning. In this paper, we propose a new method to predict final student grades. The method employs Word2Vec and Artificial Neural Network (ANN) to predict student grade in each lesson based on their comments freely written just after the lesson. In addition, we apply a window function to the predicted results obtained in consecutive lessons to keep track of each student's learning situation. The experiment results show that the prediction correct rate reached 80% by considering the predicted student grades from six consecutive lessons, and a final rate became 94% from all 15 lessons. The results illustrate that our proposed method continuously tracked student learning situation and improved prediction performance of final student grades as the lessons go by.

Keywords

PCN Method, Word2Vec, ANN, Comment Mining, Grade Prediction

1. INTRODUCTION

Learner performance assessment is a continuous and an integral part of the learning process [4]. During studying, exams are used to help teachers know how good students are learning, as well as to help them find out the difficulties with the course. However preparing a good exam is a laborious and resource demanding work, so it's still hard to obtain assessment by exams over all periods of a semester.

Thus, in the past four decades, researchers have been working on predicting individual or group performance in courses for getting assessments. By accurate predictions, we can detect students who have difficulties with the courses early, and help them improve [1].

To control students' learning behavior and situations, previous studies have used various regular assessment methods, such as e-learning logs, test marks and questionnaires. The current study proposes a new method to predict student grades. Our method is based on students' free-style comments collected after each lesson.

K.Goda, S.Hirokawa, and T.Mine [3] [2]proposed the PCN method to estimate student learning situations from free-style comments written by the students. The PCN method categorizes the comments into three items: P (Previous activity), C (Current activity), and N (Next activity).

In this paper, we apply the Word2Vec method to the comments data to get a vector representation of each comment. Then we use an artificial neural network (ANN) model to predict student grades based on the vectors. The experiments were conducted to validate the proposed methods by calculating the F-measure and accuracy for each lesson. After acquiring a prediction result for each lesson, we applied a window function and a majority vote method to get a final prediction result based on multiple lessons. The experiment results illustrate that the prediction correct rate reached 80% by considering the predicted student grades obtained from six lessons, and the final rate became 94% from all 15 lessons.

Contributions of this paper are threefold. First, we propose a new method to predict final student grades by using Word2Vec and ANN. Second, we improve the prediction performance by considering the results obtained in consecutive lessons. We show as the size of the lessons increases, the prediction performance becomes better. Third, we conduct experiments to illustrate the effectiveness of the proposed methods. The experiment results show the validity of the proposed methods.

2. RELATED WORK

Extensive literature reviews of the Educational Data Mining (EDM) research field are mainly focused on retention of students, improving institutional effectiveness, enrollment management and alumni management. In the past four decades, a considerable amount of research has gone into predicting individual or group success in exams and courses.

Schoor and Bannert [7] studied sequences of social regulatory processes (i.e. individual and collaborative activities of analyzing, planning...aspects) during collaborative sessions

and their relationship to group performance. They used process mining to identify process patterns for high versus low group performance dyads. The result models showed that there were clear parallels between high and low achieving dyads in a double loop of working on the task, monitoring, and coordinating.

Liu and Xing [5] aimed to develop a predictive model of student behavior by an ensemble approach composed of creation of sampled sets, generation of base models, and selection of base models to be aggregated for obtaining the final ensemble model. The solution required less computation resource, had satisfying prediction performance and produced prediction models with good capability of generalization.

Different from the above studies, Goda et al. [3] proposed the PCN method to estimate students' learning situations with their free-style comments written just after a lesson. They applied Support Vector Machine (SVM) to the comments for predicting final student results in 5 grades. The experiment results illustrate that as student comments get higher PCN scores, prediction performance of student grades becomes better. Sorour et al.[8] applied machine learning technique: artificial neural network (ANN) and made it learn the relationships between comments data analyzed by Latent semantic analysis(LSA) and the final student grades. They constructed a network model to each lesson. The average prediction accuracy of student final grades was 82.6%. In this study, as an extension of Sorour et al. [8], we focused on using different text mining method Word2Vec combined with the ANN model to get prediction on each lesson, and obtain prediction results based on consecutive multiple lessons. Our method outperformed the method of Sorour et al.[8].

3. METHODOLOGY

3.1 Collecting Comments

In this research, we used the same comment data as Sorour et al.[8]. The comments were collected after each lesson in a course including 15 lessons. 123 students attended this course. They were asked to fill in three simple questionnaire items about their learning status. Goda et al. [3] called the three items, P (Previous), C (Current) and N (Next) items. In this paper, we mainly focus on the C (Current) comments. Table 1 displays the real number of comments in each lesson that we analyzed. On average there is 111.13 comments in each lesson.

Table 1: Number of comments for each lesson

Lesson	Num	Lesson	Num	Lesson	Num
1	100	6	116	11	107
2	121	7	104	12	109
3	118	8	103	13	107
4	115	9	107	14	111
5	123	10	111	15	121

3.2 Comments Data Preparation

3.2.1 Comments Data Preprocessing

This step covers all the preparations required for constructing the final dataset from the initial data. Our method used a Japanese morphological analyzer Mecab[1] to analyze C comments, extract words and part of speech. In this experiment, we only used noun, verb, adjective and adverb. The number of words appeared in the comments is about 1400 in each lesson, and the number of words in all the comments without duplication is over 430 in each lesson.

3.2.2 Word2Vec

Word2vec is a popular neural network based approach to learning distributed vector representations for words released by Google in 2013. This tool adopts two main model architectures, Continuous Bag-of-Words (CBOW) and Skip-Gram[6].

3.3 Training Phase

After the previous step and before we applied ANN to train the data, we have some pretreatments for preparing training data for ANN.

We have got a list of vocabularies and their corresponding vectors after the previous step. Now we need to find out all the words one student have used in his/ her comment which existed in the vocabulary list, and add the vectors indicating these words up to get a final vector for that student.

After obtaining a list of vectors for each student, we need to proceed the training phase with the list. In this research, we used a three-layered Artificial Neural Network to estimate student grades. In our work, we used FANN Libraries[2] to build our network model. We took the results from the former step and put them into the input layer of ANN. For all the lessons, we applied the same model with 0.1 learning rate and 0.3 momentum.

3.4 Test Phase

To predict student grades, we used 5 grade categories instead of real marks to classify final student marks.

Table 2: 5 Grades Categories

Real Marks	Grades	Num of Students
≥ 90	S	21
80-89	A	41
70-79	B	23
60-69	C	17
≤ 60	D	21

Since in each lesson, there exist some students who did not fill in questionnaires, we can't predict their grade. In these cases, we treat them as grade D instead.

After training the ANN model, we proceed the test phase to get prediction results of final student grades in each lesson. In the test phase, we evaluated prediction performance (Accuracy, F-measure) by 10-fold cross validation. We separated comments data by using 90% as training data and the rest 10% as test data. The procedure was repeated 10 times and the results were averaged. Afterwards, we apply an window function and the majority vote method to obtain a continuous prediction. The details of the window function and the majority vote method will be described in Section 4.1.

[1] http://sourceforge.net/projects/mecab/

[2] http://leenissen.dk/fann/wp/

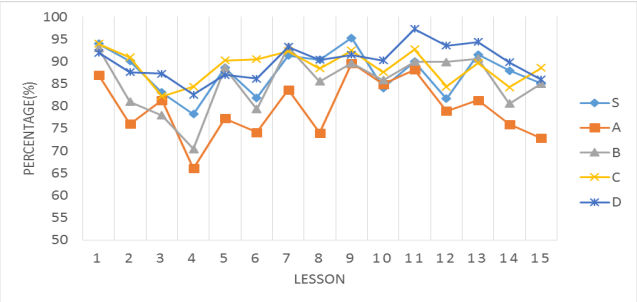


Figure 1: Accuracy for different grades

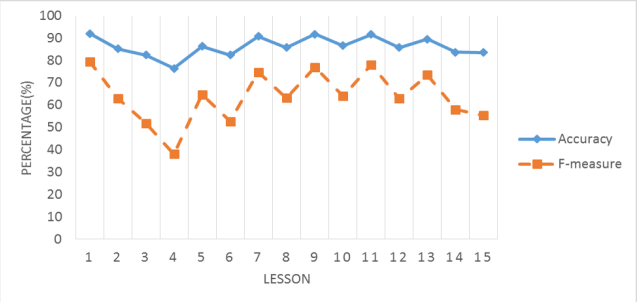


Figure 2: Average accuracy and F-measure of all the grades in each lesson

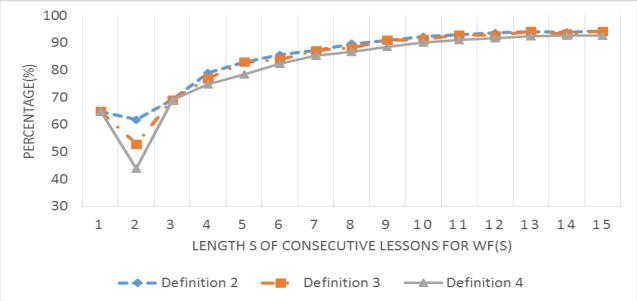


Figure 3: Average TP rate based on different definitions

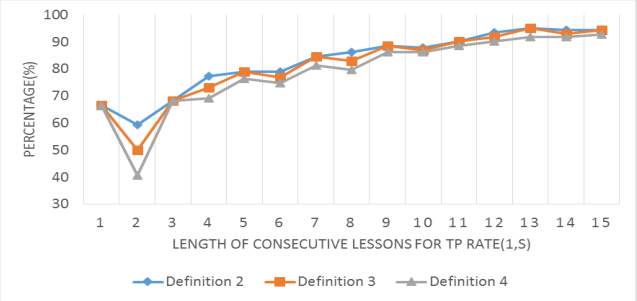


Figure 4: TP rate for different length of consecutive lessons from lesson 1

4. PREDICTION PERFORMANCE

4.1 Measure of Prediction Performance

We define the majority vote method and the window function as follows:

Let G be a set of grades $\{g_0, g_1, g_2, g_3, g_4\}$; each element of G corresponds to each grade, i.e., g_0, g_1, g_2, g_3, and g_4 correspond to S, A, B, C, and D, respectively. Let $MV_k(m,n)$ be the function of Majority Vote of student k from lessons m to n . $MV_k(m,n)$ returns a set of predicted student k's grades whose occurrence frequency from lessons m to n became the greatest. We define $MV_k(m,n)$ in Definition 1.

Definition 1. $MV_k(m,n)$

$$MV_k(m,n) = \underset{g_i \in G}{\operatorname{argmax}}\ f(k,g_i)(m,n)$$

where $f(k,g_i)(m,n)$ returns the occurrence frequency of predicted grade g_i of student k from lessons m to n.

For example, if the predicted grades of student 1 from lessons 1 to 3 are respectively S ($=g_0$), A ($=g_1$), and S ($=g_0$), then $f(1,g_0)(1,3) = 2$ and $f(1,g_1)(1,3) = 1$. So, $MV_1(1,3)$ returns $\{g_0\}$. If the predicted grades of student 1 from lessons 1 to 3 are respectively S($=g_0$), A($=g_1$), B($=g_2$), then $f(1,g_0)(1,3) = 1$, $f(1,g_1)(1,3) = 1$, and $f(1,g_2)(1,3) = 1$. So, $MV_1(1,3)$ returns $\{g_0, g_1, g_2\}$.

Function δ returns a score according to the results returned by a Majority Vote function $MV(m,n)$ defined in Definition 1. Three δ functions: δ_1, δ_2, and δ_3, are defined in Definitions 2, 3, and 4. Here we use the notation $|.|$ that denotes the cardinality of a set. For example, if $MV_1(1,3)$ returns $\{g_0, g_1, g_2\}$, then $|MV_1(1,3)| = 3$.

Definition 2. δ_1

$\delta_1(MV_k(m,n))$ returns 1 if g_k is the actual grade of student k, $g_k \in MV_k(m,n)$ and $g_l \notin MV_k(m,n)$ such that $|l-k| > 1$, 0 otherwise.

For example, we assume that the actual grade of student k is g_0, if $MV_k(m,n) = \{g_0, g_1\}$, then $\delta_1(MV_k(m,n)) = 1$. If $MV_k(m,n) = \{g_0, g_2\}$ then $\delta_1(MV_k(m,n)) = 0$, because $|2-0| > 1$.

Definition 3. δ_2

$\delta_2(MV_k(m,n))$ returns $\frac{1}{|MV_k(m,n)|}$ if $g_k \in MV_k(m,n)$ where g_k is the actual grade of student k, 0 otherwise.

Definition 4. δ_3

$\delta_3(MV_k(m,n))$ returns 1 if $g_k \in MV_k(m,n)$ and $|MV_k(m,n)| = 1$, 0 otherwise.

Next, we define $TP(m,n)$ that returns True Positive (TP) rate from lessons m to n in Defition 5.

Definition 5. $TP(m,n)$

$$TP(m,n) = \frac{\sum_{k=1}^{N_s} \delta(MV_k(m,n))}{N_s}$$

where N_s is the number of students.

Now we define function $WF(s)$, which returns the average TP rate in s consecutive lessons, in Defition 6. Here s denotes the length of consecutive lessons, i.e. the number of lessons.

Definition 6. $WF(s)$

$$WF(s) = \frac{\sum_{k=1}^{N-s+1} TP(k,k+s-1)}{N-s+1}$$

where N is the number of all lessons in a course, 15 in this research.

For example, when $N = 15$, $WF(1)$ to $WF(15)$ are computed as follows:

$$WF(1) = \frac{TP(1,1) + TP(2,2) + .. + TP(15,15)}{15}$$
$$WF(2) = \frac{TP(1,2) + TP(2,3) + .. + TP(14,15)}{14}$$
$$WF(3) = \frac{TP(1,3) + TP(2,4) + .. + TP(13,15)}{13}$$
$$\cdots$$
$$WF(14) = \frac{TP(1,14) + TP(2,15)}{2}$$
$$WF(15) = \frac{TP(1,15)}{1} = TP(1,15)$$

4.2 Results in Each Lesson

We examined the same model on all the students with different final grades. Results are shown in Figures 1 and 2. Figure 1 displays the plot of accuracy results of students with different grades in each lesson. Table 3 shows the average overall prediction accuracy and F-measure for the different grades. As for accuracy, the result of grade D is the highest, which scores 89.5%, and the lowest average is grade A, which scores 79.1%. Also, according to Figure 2, lesson 1 has the highest accuracy and F-measure, while lesson 4 has the lowest results.

Table 3: Average accuracy and F-measure for different grades

Grades	Accuracy	F-measure
S	87.3	65.6
A	79.1	**71.3**
B	85.0	62.6
C	88.5	57.2
D	**89.5**	62.3
Average of all grades	85.9	63.8

4.3 Results after Using Window Function and Majority Vote

Before we apply the window function to all the consecutive lessons, we first treat all the students who did not describe comments as Grade D. After this step, it also ensures that for each lesson, every student has one predicted grade. After we get the prediction result in each lesson, we apply the window function and the majority vote method to get a continuous track of student performance.

Here, we only consider TP rates. First we investigated the effect of size s of $WF(s)$ by varying the value of s from 1 to 15. As we can see, in Figure 3, the TP rate was increased as the value of s increased. As an example of the results, even though the strictest way of counting the correct case by Definition 4, the correct rate still raised over 80% after considering more than six lessons. In addition, with all the lessons, the correct rates all reached over 90%. And with Definition 2 and 3, they both reached 94%. The results by Definition 4 reached 92.7%.

Figure 4 shows the result of TP rate from TP(1, 1) ,TP(1, 2),TP(1, 3) to TP(1, 15) with three different definitions. With the growing of window function size, the TP rate raised over 80% with more than 7 lessons, which is slightly lower than the average.

Considering the results of Figures 3 and 4, we can say the both results took similar tendency that the TP rates became greater as the size of lessons increased.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the prediction method of student grade based on the C comments data from Goda et al. [3]. We applied the Word2Vec and ANN methods to the comments to obtain prediction of their grades in each lesson. Then we used the window function and the majority vote method to improve the prediction results based on consecutive multiple lessons. The experiment results illustrate the validity of the proposed method.

This study expressed the correlation between self-evaluation descriptive sentences written by students and their academic performance by predicting their grade. Especially when using prediction results obtained in consecutive lessons, the prediction result has quite high credibility. This could help giving feedback to students during the semester to help students achieve higher motivation and know their learning conditions better.

However, there still remain some room for improving prediction results in each lesson. In the future, we will try to apply better models to achieve higher accuracy in predicting student grades.

6. ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Numbers 25350311 and 26540183.

7. REFERENCES

[1] M. Fire, G. Katz, Y. Elovici, B. Shapira, and L. Rokach. Predicting student exam's scores by analyzing social network data. Lecture Notes in Computer Science Volume 7669, pp 584-595, ,2012.

[2] K.Goda, S.Hirokawa, and T.Mine. Correlation of grade prediction performance and validity of self evaluation comments. In Proc. of the 14th annual ACM SIGITE conference on Information technology education, Florida, USA, 2013.

[3] K.Goda and T.Mine. Analysis of student' learning activities through qualifying time-series comments. Proc. of the KES 2011, Part 2, LNAI 6882, Springer-Verlag Berlin Heidelberg, pp.154-164, 2011.

[4] L.Earl. Assessment of Learning, for Learning, and as Learning. Thouand Oaks,CA,Corwin Press, 2003.

[5] K. Liu and Y. Xing. A lightweight solution to the educational data mining challenge. In Proceedings of the KDD 2010 cup 2010 workshop knowledge discovery in educational data(pp. 76-82), 2010.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. Advances in Neural Information Processing Systems (NIPS), 2013.

[7] C. Schoor and M. Bannert. Exploring regulatory processes during a computer-supported collaborative learning task using process mining. Computers in Human Behavior, 2012.

[8] S. E. Sorour, T. Mine, K. G., and S. Hirokawa. Predictive model to evaluate student performance. In Journal of Information Processing (JIP), Vol. 23, 2015.

Learning the Creative Potential of Students by Mining a Word Association Task

Cristian Olivares-Rodríguez
Engineering Faculty, Universidad Andres Bello
Avenida República 237
Santiago 8370146, Chile
colivares@unab.cl

Mariluz Guenaga
DeustoTech - Deusto Institute of Technology -
University of Deusto
Avenida Universidades 24
Bilbao 48007, Spain
mlguenaga@deusto.es

ABSTRACT

Creativity is a relevant skill for human beings in order to overcome complex problems and reach novel solutions based on unexpected associations of concepts. Thus, the education of creativity becomes relevant, but there are not tools to automatically track the creative potential of learners over time. This work provides a novel set of behavioural features about creativity based on associative skills. These associations are processed to define two models that depict students' creative potential. This way, we have reached an acceptable accuracy rate in the classification of creative potential, hence we have found concrete evidence regarding the ability to automatically predict the creative potential of students based on their association capabitilies.

1. INTRODUCTION

Creativity generally emerges when people face a problematic or new situation, where constraints and concepts are probably unknown. An intensive search of novel solutions is required to solve real problems. This search can be done by exploration, transformation or combination of concepts. Therefore, there is a need of new associations of concepts to reach unknown solutions [7].

Nowadays, students are the centre of their learning when solving authentic problems, and creative skills provide students adaptation abilities to overcome heuristic environment, where nor the path to the solution nor the solution are known and therefore, you need to establish strategies to achieve the goal. In this context, the intensive use of technology by students to produce web searches and social data is a rich source of information to learn how students behave [2], thus a monitoring framework of creativity, based on associative features, becomes feasible.

A set of creative challenges have been applied to 64 students of sixth grade in two primary schools in Spain. First, we have applied an "unusual uses" test as a measure of creativity. After that, we have applied two "word association" tasks in order to depict their associative skills over time, similarly to [4]. Based on data captured from these activities we have extracted a set of relevant features regarding to creativity in order to model the user behaviour.

Our hypothesis bases on the fact that the local frequency of words and the time when they came out provide relevant information about originality. Also, we set that time provides a measure about fluency and that part of speech gives information about flexibility.

We have tested the strength of features associated to creativity with a supervised classification approach. We propose a model to track the creative potential of students based on their associative skills, but it still requires a more powerful set of semantic features and a learning algorithm that works properly with sequential data. However, the developed model provides an acceptable accuracy rate (over 81% in the best case) and outperform a Bag-of-Words approach.

(Sec-2) describes the concept of creativity and provides formal definitions of existing models (Sec-3). (Sec-4) defines the experiment carried out to collect data and evaluates the predictability of proposed models. Then, we present the main results using these models (Sec-5) and provide a discussion regarding the monitoring of creativity (Sec-6). Finally, we summarize our contributions and outline future works (Sec-7).

2. BACKGROUND

Creativity is a mental process based on associations in our mind and it has been characterized by: fluency, flexibility, originality and elaboration. The conceptual model of creativity of Amabile [1] defines a general process to solve problems that is grouped in three phases: a *conceptualization* phase to establish several problem definitions; a *search* phase to reach concepts, make new associations and establish new solutions; and a *development* phase to implement a solution and to update the knowledge. The model also introduces relevant skills related to creativity: associative and executive, which have been studied against the creative performance [4].

Mednick [7] proposed an associative theory of creativity where affirms that a creative person is able to find new solutions to real problems making as many associations as possible

(fluency), as diverse as possible (flexibility) and as unexpected as possible (originality). Benedek et al. have found a positive relation between fluency and creativity [4].

There are some procedures to measure the creativity [8] [4]. Generally, it is measured by an *unusual uses* test, where each participant must achieve as many uses as possible for a particular object (e.g. a brick) in a short period of time, and the expert evaluation of creativity is based on a Likert-scale of fluency, flexibility and originality. This methods do not include measures of the creative potential obtained during data from the process to carry out the activity (i.e. search on the Web, social networks, etc.) and, thus there is an opportunity to model the creativity to implicitly depict students' behaviour.

A *word association* task makes visible the association skills by retrieving as many words as possible with respect to a query word, in a short period of time. This task is an heuristic process of word retrieval, where a user defines an association model Q^u in order to provide words w_i, in a certain time t_i and related to a query word q^s (Eq-1).

$$Q^u(q^s) = [(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)] \forall w_i \in W^u \quad (1)$$

Moreover, each user defines an heuristic measure h_u^* based on a hidden similarity measure S^u (Eq-2).

$$h_u^*(q^s, w) = S^u(q_j^s, w|t) \mid u \in U \quad (2)$$

Even heuristic is hidden, we can derive an empirical model through a set of association features [4].

2.1 Computational Models of Creativity

In art, a search behaviour analysis based on a visual creative task has been developed to figure out the hidden process [5]. A computerized aesthetic composition task was implemented in order to capture the search flow followed by each participant to design a new image. Thus, the user actions are used to depict the heuristic applied in the search process.

In education of creativity, a personalized creativity learning system (PCLS) based on decision trees has been proposed [6]. Nevertheless, they are not focused in track the creativity, but in enhance the process to teach creativity. The purpose of the PCLS is to adapt the student path based on a set of creativity measures and demographic information (gender, college, etc.).

3. MODEL OF WORD ASSOCIATIONS

We have defined two user models: the Bag-of-words (U_{BoG}) and the Features of Creativity (U_{FoC}).

3.1 Bag-of-Words for Association Tasks

The bag-of-words (BoW) is a basic model to describe the content of documents in the information retrieval domain (Eq-3).

$$BoG(d_j) = (f(w_1), f(w_2), \ldots f(w_n)) \forall w_i \in W \quad (3)$$

Where d_j is a document, $f(w_i)$ is a function that defines the relation of the word w_i with the document d_j and w_i is in the dictionary W. This model provides a measure of the originality of words. A document d_j can be defined as the set of all associated words that users have provided against a query word q^s (Eq-4).

$$d_j(q^s) = \bigcup_{k=1}^{|U|} Q^k(q^s) \quad (4)$$

And, the word frequency wf (Eq-5) is defined as an originality measure of the word regarding each document and a relative time measure [3].

$$wf(w_i) = \sum_{j=1}^{|D|} \left[\frac{f(w_i, d_j)}{\max\{f(w, d_j) | w \in d_j\}} \times \frac{t_i}{|T|} \right] \quad (5)$$

We define a model of the user based on BoW (Eq-6), where $\overline{W}^u$ is the set of all words provided by the user u.

$$U_{BoW} = \bigcup_{i=1}^{|W|} \begin{cases} wf(w_i) & , w \in \overline{W}^u \\ 0 & , otherwise \end{cases} \quad (6)$$

3.2 Features of Creativity

We measure *Fluency* as the time variance of each query word of the user as you can see in the Eq-7, where T^u is the set of timestamp of each answer of the user u to the query word q^s.

$$t_v^u(q^s) = var[(t_1), (t_2), \ldots, (t_n)] \forall t_i \in T^u \quad (7)$$

We measure *Flexibility* as the variance in the Part of Speech (PoS) of associated words (Eq-8), where W^u are the answers of the user u to the query word q^s.

$$PoS_v^u(q^s) = var[PoS(w_1), \ldots, PoS(w_n)] \forall w_i \in W^u \quad (8)$$

We define *Originality* through two features based on the word frequency: 1) the variance of the word frequency (Eq-9), where W^u is the set of answers of the user u to the query word q^s.

$$wf_v^u(q^s) = var[f(w_1, d_j), \ldots, f(w_n, d_j)] \forall w_i \in W^u \quad (9)$$

And 2) the dot product of frequency and time (Eq-10)

$$t.f^u(q^s) = [\frac{t_1}{|T|}, \ldots, \frac{t_n}{|T|}] \cdot [f(w_1, d_j) \ldots f(w_n, d_j)] \quad (10)$$

Hence, we define a feature vector $fv^u(q^s)$ integrating the Equations 7, 8, 9 and 10, as follows:

$$fv^u(q^s) = [t_v^u(q^s), PoS_v^u(q^s), wf_v^u(q^s), t.f^u(q^s)] \quad (11)$$

Finally, we model the user behaviour U_{FoC} (Eq-12) concatenating the feature vector fv^u of each association task (Eq-11), where $|Q|$ is the set of all association tasks driven by the query words q_i^s.

$$U_{FoC} = \bigcup_{i=1}^{|Q|} fv^u(q_i^s) \quad (12)$$

4. EXPERIMENTAL SETUP

This work aims to model the hidden heuristic in association tasks based on the behaviour of creative people. We have designed a practical experiment based on a Web platform to generate a novel dataset that relates associative skills of users and their creative potential. We applied this experiment on sixth grade students from two different schools in Spain, in a relation of 67% from one school and 33% from the other. The whole sample was composed by 47% of male and 53% of female.

The experiment involved two creative challenges developed during a class: *unusual uses* and *word association* tasks. First, the users were asked to: *write down as many unusual uses as possible for the object 'Shoe' during 60 seconds.* With this task we captured data about the divergent thinking potential of each user and, thus, we can compute a measure of their creative potential. The users were also asked to: *write down as many associated words as possible for a 'Book' ('Door') during 60 seconds.*

In order to form the dataset, the platform has registered the query object, the unusual uses listed by students and the timestamp for each use. It also has saved the query words, the associated words provided by students and the timestamp of each word. These data is represented by equation 1. Demographic data collected from each student includes age, gender and country.

In addition, a label about their creativity was provided based on the unusual uses challenge and the intrinsic characteristics of creativity. Two reviewers labelled each user as creative or non-creative using a Linkert-scale (5) of flexibility, fluency and originality. Accordingly, a labelled dataset was defined to perform a supervised learning of the creative behaviour of users. The dataset structure is depicted in the Table 1.

We have modelled the user behaviour (Sec-3) using the modified Bag-of-Words (BoW) and the Feature of Creativity (FoC). We have designed a two-class supervised learning

Table 1: User information in dataset

Attribute	Description
`Gender`	The user gender
`Age`	The user age
`Country`	The user country
`Creative tag`	Creative (+) or No creative (-)
`Unusual Uses`	A set of tuples *(use, time)* per object
`Associations`	A set of tuples *(word, time)* per query

Table 2: Dataset statistics

Avg. Attr. per minute	Creative (53%)		No Creative (47%)	
	M (35%)	F (65%)	M (60%)	F (40%)
#Uses 'Shoe'	3.33	4.20	**4.88**	4.10
#Asoc 'Book'	**6.67**	**6.05**	5.30	5.73
#Asoc 'Door'	**5.92**	**5.77**	4.5	4.18
	Age of Students			
	11	12	11	12
#Uses 'Shoe'	3.94	3.33	3.91	4.00
#Asoc 'Book'	**6.42**	4.67	5.52	5.33
#Asoc 'Door'	**5.93**	4.67	4.72	5.33
Diccionary Size (# unique words)				
Global	'Libro'		'Puerta'	
247	127		129	

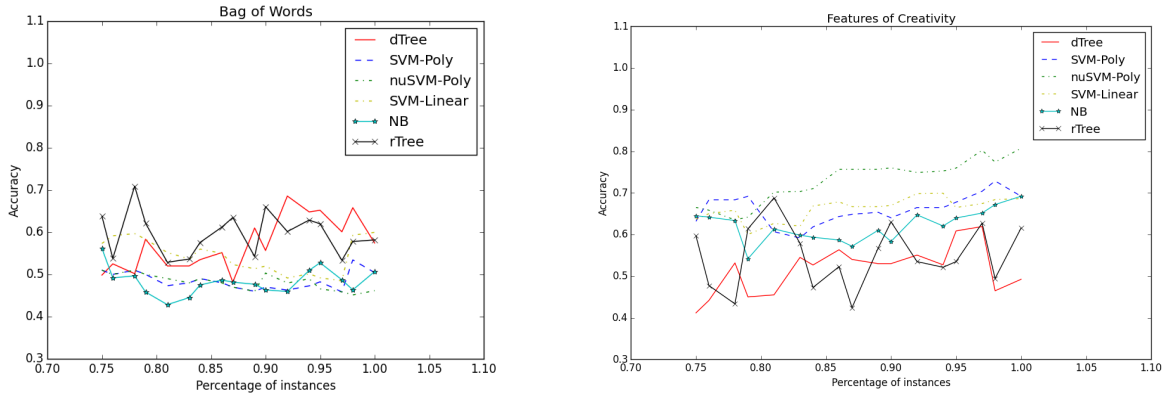
experiment and trained a set of learning algorithms: Naïve Bayes (NB), Decision Tree ($dTree$), Support Vector Machine (three kernels) and Random Forest ($rTree$). In order to evaluate the accuracy of the learning algorithms we performed a cross-validation method. Thus, we have iteratively divided the dataset in k subsets, where the $k-1$ subsets were used to train the algorithms and the last one was used to validate the prediction quality based on its accuracy. Finally, we have performed an analysis of accuracy results against the percentage of the instances used in the cross-validation method.

5. RESULTS

By applying the challenges, a dataset was defined based on Eq-1. We highlight that creative students are more fluent than non-creative ones and younger students provide more associated words per minute. We have also defined a global dictionary with all associated words W provided by users and local dictionaries (W^{q^s}) for each association task. A more detailed information is shown in the Table 2

We have analysed the size of the dataset, because the features are based on statistics. In the figure 1a you can see the accuracy of the U_{BoW} model, which approximately ranges in 10 points at each model. The results of the model are similar for different sizes of the dataset, so this model can be seen independent of the size of the dataset. In the figure 1b we show the accuracy of the U_{FoC} model, which generally increases with respect to the size of the dataset, except in the case of the tree-based method. This model can be seen as dependent of the dataset size and it should improve as the dataset grows.

The most stable algorithms are the kernel-based (SVM) because they fit more precisely with the features of creativity.



(a) (b)

Figure 1: Accuracy performance against different sizes of dataset: The strength of a) U_{BoW} and b) U_{FoC}.

Also, we reach high levels of accuracy in the classification of creative behaviour based on a simple set of features and a moderate number of available samples, which reach up to 81% in the U_{FoC}.

6. DISCUSSION

The work by Jennings et al. is a not context free proposal (art) and, it is too invasive for students [5]. The associative actions of users are mined to figure out the hidden strategy of users through a design task. We define a model based on a ordinary task of word association that is common in problem-solving contexts, web searches and social networks. Therefore, the provided models can be applied in active learning contexts where students make associations. The proposal of Lin et al. [6] is seeking to improve the learning of creativity by recommendation in a personalized tutoring system. We propose a complementary work to identify the creative potential and, thus, it could be possible to provide better learning paths to students based on such prediction.

The bag-of-words approach U_{BoW} has reached an acceptable accuracy level, but it has a high variance. The features of creativity approach U_{FoC} is more stable and it has a growing accuracy along the number of samples. This model is based on a small number of features, which are highly related with the theoretical features of creativity: fluency, frequency and originality.

7. CONCLUSION

We have proposed two user models to identify creative students when they associate words: U_{BoW} and U_{FoC}. These models outline the creativity of students over time by exploiting their word associations (Web search, Social Network, etc). Thus, we depicted that it is possible to learn a classifier based on associative features with an acceptable accuracy. We have developed a dataset that relates the association skills and the creative potential of students.

In the future we will integrate the sequentially of associations, so there is the possibility to use sequential learning algorithms. The flexibility could be described using the sense/meaning of the word as a more informative similarity. Finally, we have depicted that a higher number of instances improves performance, then a more diverse set of samples should be considered.

8. ACKNOWLEDGMENTS

This work is partially founded by the Erasmus Mundus SUDUE program.

9. REFERENCES

[1] T. Amabile. The social psychology of creativity: A componential conceptualization. In *Journal of Personality and Social Psychology*, volume 45, pages 357–376, 1983.

[2] P. André, M. c. Schraefel, J. Teevan, and S. D. T. Discovery is never by chance: designing for (un) serendipity. *C and C '09*, pages 305–314, 2009.

[3] R. E. Beaty and P. J. Silvia. Why do ideas get more creative across time? an executive interpretation of the serial order effect in divergent thinking tasks. *Psychology of Aesthetics, Creativity, and the Arts*, 6(4):309–319.

[4] M. Benedek and A. C. Neubauer. Revisiting mednick's model on creativity-related differences in associative hierarchies. evidence for a common path to uncommon thought. *The Journal of Creative Behavior*, 47(4):273–289, 2013.

[5] K. E. Jennings, D. K. Simonton, and S. E. Palmer. Understanding exploratory creativity in a visual domain. In *Proceedings of the 8th ACM conference on Creativity and cognition*, pages 223–232. ACM, 2011.

[6] C. F. Lin, Y.-C. Yeh, Y. H. Hung, and R. I. Chang. Data mining for providing a personalized learning path in creativity: An application of decision trees. *Comput. Educ.*, 68:199–210, Oct. 2013.

[7] S. Mednick. The associative basis of the creative process. In *Psychological review*, volume 69, pages 220–232, 1962.

[8] E. P. Torrance. The nature of creativity as manifest in its testing. In *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge University Press, New York, 1988.

Optimizing Partial Credit Algorithms to Predict Student Performance

Korinn Ostrow, Christopher Donnelly, Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
{ksostrow, cdonnelly, nth}@wpi.edu

ABSTRACT

As adaptive tutoring systems grow increasingly popular for the completion of classwork and homework, it is crucial to assess the manner in which students are scored within these platforms. The majority of systems, including ASSISTments, return the binary correctness of a student's first attempt at solving each problem. Yet for many teachers, partial credit is a valuable practice when common wrong answers, especially in the presence of effort, deserve acknowledgement. We present a grid search to analyze 441 partial credit models within ASSISTments in an attempt to optimize per unit penalization weights for hints and attempts. For each model, algorithmically determined partial credit scores are used to bin problem performance, using partial credit to predict binary correctness on the next question. An optimal range for penalization is discussed and limitations are considered.

Keywords

Partial Credit, Student Modeling, Next Question Correctness, Adaptive Tutoring Systems, Maximum Likelihood, Grid Search

1. INTRODUCTION

Adaptive tutoring systems provide rich feedback and an interactive learning environment in which students can excel, while teachers maintain data-driven classrooms by using the systems as powerful assessment tools. Simultaneously, these platforms have opened the door for researchers conducting minimally invasive educational research at scale while offering new opportunities for student modeling. Still, they are commonly restricted to measuring performance through binary correctness on each problem. Arguably the most popular form of student modeling within computerized learning environments, Knowledge Tracing, is rooted in the binary correctness of each opportunity or problem a student experiences within a given skill [1]. Knowledge Tracing (KT) drives the mastery-learning component of renowned tutoring systems including the Cognitive Tutor series, allowing for real time predictions of student knowledge, skill mastery, or next problem correctness [4]. Similar modeling methods consider variables that extend beyond correctness but rarely escape the binary nature of the construct, including Item Response Theory [2] and Performance Factors Analysis [9]. By restricting input to a binary metric across questions, these modeling techniques fail to consider a continuous metric that is commonplace for many teachers: partial credit.

Partial credit scoring used within adaptive tutoring systems could provide more individualized prediction and thus establish models with better fit. It is likely that binary correctness has remained the default for learning models due to the inherent difficulty of defining a universal algorithm to generalize partial credit scoring across platforms. Some of the onus may also fall on users' familiarity with current system protocol; students tend to avoid using system feedback regardless of the benefits it may provide because requesting feedback results in score penalization. However, the primary goal of these platforms is generally to promote student learning rather than simply acting as an assessment tool, and thus, binary correctness is flawed.

The present study considers data from ASSISTments, an online adaptive tutoring system that provides assistance and assessment to over 50,000 users around the world as a free service of Worcester Polytechnic Institute. Researchers have previously used ASSISTments data to modify student-modeling techniques in a variety of ways including student level individualization [7], item level individualization [8], and the sequence of student response attempts [3]. Previous work has also shown that naïve algorithms and maximum likelihood tabling methods that consider hints and attempts to predict next problem correctness can be successful in establishing partial credit models meant to supplement KT [10; 11]. More recently, algorithmically derived partial credit scoring resulted in stand-alone tabled models using data from only the most recent question and yet showing goodness of fit measures on par with KT at lower processing costs [6]. However, we hypothesize that some conceptualizations of partial credit may lead to better predictive models than others. Rather than subjectively defining tables or algorithms, a data driven approach should be considered. Thus, considering student performance within the ASSISTments platform, the current study employs a grid search on per unit penalizations of hints and attempts to ask:

1. Based on penalties for hints and attempts dealt per unit, is it possible to algorithmically define partial credit scoring that optimizes the prediction of next problem correctness?
2. Does the optimal model of partial credit differ across different granularities of dataset analysis?

Establishing an optimal partial credit metric within ASSISTments would allow teachers using the tool to more accurately assess student knowledge and learning, while allowing students to alter their approach to system usage by taking advantage of adaptive feedback. The optimization of partial credit scoring would also enhance student modeling techniques and offer a new approach to answering complex questions within the domain of educational data mining.

2. DATA

The ASSISTments dataset used for the present study is comprised solely of assignments known as Skill Builders. This type of assignment requires students to correctly answer three consecutive questions to complete the problem set. Questions are randomly pulled from a large pool of skill content and are typically presented with tutoring feedback, most commonly in the form of hints. The dataset has been de-identified and is available at [5] for further investigation.

The dataset used in the present study is a compilation of Skill Builders from the 2012-2013 school year, containing data for 866,862 solved problems. Recorded data includes students' performance on the problem (i.e., binary correctness, hint count, attempt count), variables that identify the problem itself (i.e., problem type, unique problem identification number) and information pertaining to the assignment housing the problem (i.e., unique identifiers for assignments, skill type, teachers, and schools). The dataset was representative of 120 unique skills and 24,912 unique problems, solved by 20,206 students.

On average, students made 1.53 attempts per problem (SD = 15.08). The minimum number of attempts was 0 (i.e., a student who opened the problem and then left the tutor), while the maximum number of attempts was a daunting 12,246 (i.e., a student who hit 'Enter' repeatedly for a prolonged period of time, likely out of frustration or boredom). Students made a total of 1,324,226 attempts across all problems. The majority of problems (74.9%) had just one logged attempt per student (typically correct answers), while 15.1% of problems carried only two logged attempts.

Hint usage among all students averaged 0.61 hints per problem (SD = 1.29). The minimum number of hints used was 0 (i.e., no feedback requested), while the maximum number of hints used was 10. Interestingly, the maximum number of hints available for any particular problem was 7. Thus, a handful of students who logged more than 7 hints were accessing the tutor in multiple browser windows (i.e., cheating). On average there were 3.22 hints available per problem (SD = 0.89). The majority of problems contained 3 hints (44.6%), 4 hints (28.9%), or 2 hints (18.2%). Although there were 2,768,299 hints available across all problems, students only used 529,394 hints, or approximately 19% of available feedback. Bottom out hints, or those providing the problem's solution, were only used on 146,742 (16.9%) of problems.

Additional analyses were performed on the 261,787 problems that students answered incorrectly out of the original 866,862 problems solved. Within this subset of data, students made an average of 2.75 attempts per problem (SD = 27.40). Students also used an average of 2.02 hints (SD = 1.63). This subset of problems had 860,131 total hints available, of which students used 528,644 hints (61.5%).

Hint usage would likely increase if partial credit scoring was implemented within the ASSISTments platform. In many classrooms, binary first attempt scoring has created an environment in which students are afraid to use hints although they would benefit from feedback, as they know they will receive no credit. Further, the dataset suggests that once students are marked wrong, they are more likely to jump through all available hints and seek out the answer (56% of incorrect first attempts led to bottom out hinting). This reflects another substantial downfall in the system's current protocol: once the risk has passed, so has the drive to learn. The implementation of partial credit scoring has the potential to alleviate this misuse.

3. METHODS

The present study presents an extensive grid search of potential per hint and per attempt penalizations. The full dataset was used to define partial credit scores algorithmically based on per unit penalizations ranging from 0 to 1 in increments of 0.05 for both hints and attempts. Thus, for each solved problem in the dataset, 441 partial credit scores were established based on each possible combination of per unit penalization. For example, in a model in which each attempt earned a penalization of 0.05, and each hint earned a penalization of 0.1, a student who made three attempts and used one hint would receive a penalty of 0.25 ((3x0.05) + (1x0.1)), effectively scoring 0.75 on that problem. This process was used to score each problem in the dataset for each possible penalty combination, with a floored per problem score of 0 (students could not receive negative scores). This method was similar to that presented by Wang & Heffernan in the Assistance Model [10] which established a tabling method to calculate probabilities of next problem correctness based on combinations of hints and attempts that resulted in twelve possible bins or parameters.

For each of the 441 partial credit models, a maximum likelihood tabling method was employed using five fold cross validation. Within each model, a modulo operation was used on each student's unique identification number to assign students to one of five folds. Note that this method resulted in folds that all represented approximately 20% of students in the dataset. Maximum likelihood probabilities for next problem correctness were then calculated for each partial credit score within each model. Table 1 presents an average of test fold probabilities for the model in which each attempt and each hint are penalized 0.1. For instance, a student using two attempts (2 x 0.1) and one hint (1 x 0.1) would be penalized 0.3, thus falling into the score bin of 0.7 (PC Score). Following through with this example, based on 11,174 problems solved that fit this scoring structure, the average of known binary performance on the following problem was 0.599. This value becomes the prediction for next problem correctness for students scoring 0.7 on the current problem.

Using the maximum likelihood probabilities for next problem correctness within each test fold as predicted values, residuals were then calculated by subtracting predictions directly from actual next problem binary correctness (i.e., 1 – 0.725 = 0.275; 0 – 0.571 = -0.571). This approach was used rather than selecting an arbitrary cutoff point to classify a prediction as correct or incorrect in the binary sense (i.e., values greater than or equal to 0.6 serve as predictions of correctness) because it reduced the potential for researcher bias.

Table 1. Probabilities averaged across test folds for the model in which the penalization per hint and per attempt is 0.1

PC Score	n	Max. Likelihood NPC
0	149,504	0.467
0.1	422	0.571
0.2	685	0.581
0.3	1,055	0.578
0.4	1,784	0.574
0.5	3,442	0.583
0.6	6,623	0.585
0.7	11,174	0.599
0.8	18,679	0.662
0.9	49,972	0.725
1.0	476,523	0.802

4. RESULTS

For each model, residuals were used to calculate RMSE, R^2 & AUC at three levels of granularity: problem level, student level, and skill level. Heat maps are only presented here for RMSE, as the other metrics established almost identical maps. Metrics representing greater model fit are depicted using the purple end of the spectrum, while those representing poorer fit are represented using the red end of the spectrum. Further, a series of ANOVAs were conducted to compare each set of models within the same penalization level for attempts and hints. For example, the 21 models in which attempt penalty was set to 0.2 were compared to all other sets of attempt penalty models to investigate significant differences across penalties. This method was used rather than comparing each model with all other models using paired samples t-tests, as the resulting 194,481 analyses (441^2) would greatly inflate the rate of Type I error without unrealistic corrections.

Initial analysis was performed at the problem level; residuals were calculated for each problem that contained next problem correctness metrics and goodness of fit measures were averaged across the dataset. Each metric followed a similar structure in which low attempt penalties appear to result in better fitting models, while hint penalty does not appear to be significant. Thus, partial credit scoring algorithms using lower penalties for attempts were better at predicting next problem performance, as depicted in Figure 1. The ANOVA results depicted in Table 2 suggest that differences in attempt penalty models were significant. Thus, the set of models with per attempt penalties of 0.1 differed significantly from the set of models with per attempt penalties of 0.8. Differences among hint penalty models were not reliably significant. Figure 1 also suggests that the current binary scoring protocol used by ASSISTments results in predictive models that are inadequate. First attempt binary correctness is the equivalent of the model in which per attempt and per hint penalty are both set to 1, or the upper right corner of each heatmap). This model resulted in consistently poor fit metrics, suggesting that modeling techniques such as KT should employ continuous or binned partial credit values as input as they enhance next problem prediction ability. It has not yet been investigated how this alteration would change the prediction of other variables commonly predicted through KT, such as latent student knowledge or skill mastery.

Student level analysis was undertaken using a subset of the original data file. At this granularity, goodness of fit metrics were calculated for each student and averaged across students to obtain final metrics for each of the 441 models. As the ASSISTments system measures completion of a Skill Builder as three consecutive correct answers, a number of high performing students had limited opportunity counts within skills. For students with too few data points, it was not possible to calculate R^2 and AUC. Therefore, student level analysis incorporated 7,429 students from the original dataset, or 651,849 problem logs. Answering our second research question, it appears as though the region of optimal partial credit values observed at the problem level remains consistent at the student level, as shown in Figure 2. ANOVA results depicted in Table 2 show reliably significant differences across attempt penalty models but not across hint penalty models.

Skill level analysis was also undertaken using a subset of the original data file. One skill did not have enough data based on a low number of users and high mastery within those users, and was

Table 2. ANOVA results for groups of attempt and hint penalty models at each level of analysis

			Attempt Penalty			Hint Penalty		
Level	*Min*	*Max*	*F*	*p*	R^2	*F*	*p*	R^2
Problem								
RMSE	.430	.435	302.70	.000	.935	0.95	.519	.043
AUC	.626	.655	295.46	.000	.934	1.14	.304	.052
R^2	.070	.091	304.34	.000	.935	0.95	.525	.043
Student								
RMSE	.424	.429	222.49	.000	.914	1.34	.149	.060
AUC	.578	.593	208.19	.000	.908	1.42	.106	.063
R^2	.096	.110	374.52	.000	.947	0.80	.715	.037
Skill								
RMSE	.423	.429	517.85	.000	.961	0.55	.944	.026
AUC	.624	.647	250.17	.000	.923	0.72	.805	.033
R^2	.073	.090	510.96	.000	.961	0.49	.971	.023

Note. For all models, df = (20, 420).

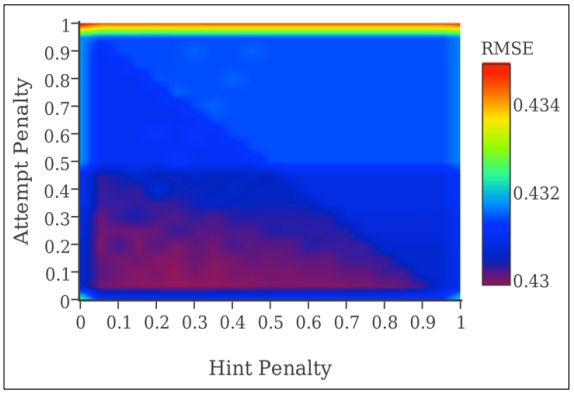


Figure 1. Problem Level RMSE

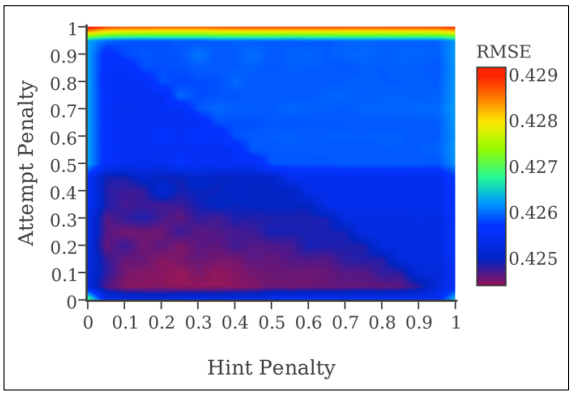


Figure 2. Student Level RMSE

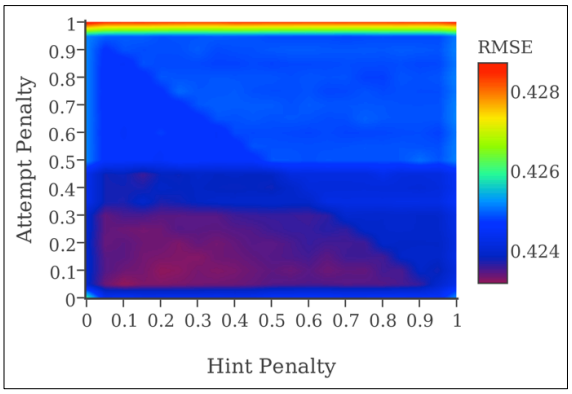


Figure 3. Skill Level RMSE

excluded from skill level analysis, resulting in a file with 119 skills. At this granularity, goodness of fit metrics were calculated for each skill and averaged across all skills to obtain final metrics for each of the 441 models. Results are depicted in Figure 3. The heat map shows that the region of optimal penalization has grown more concise, showing optimal fit among models with low per hint and per attempt penalties (< 0.3). ANOVA results depicted in Table 2 again suggest reliably significant differences in all metrics across attempt penalty models but not across hint penalty models.

Post-hoc analyses were conducted on ANOVA results using multiple comparisons to examine significant differences between attempt penalty and hint penalty model groups when considering problem level AUC. Using a Bonferroni correction to reduce Type I error, this process resulted in a series of significance estimates for penalty group comparisons (i.e., all models where attempt penalty is 0.1 compared to all models where attempt penalty is 0.3 results in a non-significant difference, $p = 0.88$). Results suggested that models close in penalty were less likely to differ significantly than models with greater difference in penalty. For instance, models with an attempt penalty of 0.1 were significantly different than those with an attempt penalty of 0.4, but were not significantly different than those with an attempt penalty of 0.2. This information can be used to help optimize partial credit penalizations, as it may be more motivating and productive for students to receive smaller penalizations. Such information could also allow systems like ASSISTments to define a range of possible penalizations that could then be refined by the teacher, providing all users with a greater sense of control.

5. DISCUSSION & CONTRIBUTION

The initial findings of a grid search on partial credit penalization through per unit hint and attempt docking suggest that the implementation of partial credit within adaptive tutoring systems can be established using a data driven approach that will ultimately produce stronger predictive models of student performance while enhancing the way adaptive tutoring systems are used by students and teachers.

Our first research question was answered with a resounding "Yes," certain algorithmically derived combinations of partial credit penalization are better than others when used to predict next problem performance. Optimal partial credit models were visible in heat maps spanning three levels of data granularity and remained relatively consistent across granularities, thus answering our second research question. ANOVAs revealed that differences in attempt penalty models were consistently significant across dataset granularities, while differences in hint penalty models were not reliable. This finding is likely due to the fact that hint usage is lower and less distributed than attempt count across problems in the dataset, and it is possible that this finding would diminish in a system that more readily promoted the use of tutoring feedback without penalization, or a system already employing partial credit scoring.

The partial credit models that we define here as optimal, based on their ability to predict next problem performance, were models with per hint and per attempt penalties of 0.3 or less. Additional analyses revealed that at the problem level, there should be no reliable difference in predictive ability of a model penalizing 0.3 per attempt from a model penalizing 0.1 per attempt, with variable hint penalization. This finding suggests that less penalization is just as effective, offering an opportunity to consider student motivation and affect when defining a partial credit algorithm. This grid search also revealed that partial credit metrics outperform binary metrics when predicting next problem performance, as previously shown in [6]. Thus, it is possible to improve prediction of student performance within adaptive tutoring systems simply by implementing partial credit scoring. It should also be noted that a leading limitation of the approach presented here is that we have only been predicting next problem correctness, rather than latent variables such as skill mastery or student knowledge. It is possible that optimizing partial credit would also provide benefits for the prediction of latent effects, but further research is necessary in this domain.

6. ACKNOWLEDGMENTS

We acknowledge funding from NSF (1316736, 1252297, 1109483, 1031398, 0742503, 1440753), ONR's "STEM Grand Challenges," and IES (R305A120125, R305C100024). Thanks to S.O. & L.P.B.O.

7. REFERENCES

[1] Corbett, A.T. & Anderson, J.R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4: 253-278.

[2] Drasgow, F. & Hulin, C.L. (1990). Item response theory. In M.D. Dunnette & L.M. Hough (Eds.), Handbook of Industrial and Organizational Psychology, Vol. 1, pp 577-636. Palo Alto, CA: Consulting Psychologists Press.

[3] Duong, H.D., Zhu, L., Wang, Y., & Heffernan, N.T. (2013). A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for a Hint. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th Int Conf on EDM. pp. 316-317.

[4] Koedinger, K.R. & Corbett, A.T. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61-78). NY: Cambridge University Press.

[5] Ostrow, K. (2014). Optimizing Partial Credit Data. Accessed 12/8/14. https://tiny.cc/OptimizingPartialCredit

[6] Ostrow, K., Donnelly, C., Adjei, S. & Heffernan, N. (2015). Improving Student Modeling Through Partial Credit and Problem Difficulty. In Russell, Woolf & Kiczales (Eds.), Proceedings of the 2nd ACM Conf on L@S. pp. 11-20.

[7] Pardos, Z.A. & Heffernan, N.T. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Proceedings of the 18th Int Conf on UMAP. pp. 255-266.

[8] Pardos, Z.A., & Heffernan, N.T. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Joseph A. Konstan et al. (Eds.): UMAP 2011, LNCS 6787, pp. 243-254.

[9] Pavlik, P.I., Cen, H., Koedinger, K.R. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In: Proceedings of the 14th Int Conf on AIED, pp. 531-538.

[10] Wang, Y. & Heffernan, N.T. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. The 24th International FLAIRS Conference.

[11] Wang, Y. & Heffernan, N. (2013). Extending Knowledge Tracing to Allow Partial Credit: Using Continuous versus Binary Nodes. In K. Yacef et al. (Eds.) AIED 2013, LNAI 7926, pp 181-188.

Identifying Styles and Paths toward Success in MOOCs

Kshitij Sharma[1], Patrick Jermann[2], Pierre Dillenbourg[1]
1. Computer Human Interaction in Learning and Instruction
2. Center for Digital Education, École Polytechnique Fédérale de Lausanne, Switzerland
<firstname>.<lastname>@epfl.ch

ABSTRACT

Current schemes to categorise MOOC students result from a single view on the population which either contains the engagement of the students or demographics or self reported motivation. We propose a new hierarchical student categorisation, which uses common online activities capturing both engagement and achievement of MOOC students. A first level is based on the online engagement with the course structure, i.e., whether they take part in graded activities or not. Based on this criterion, we divide students into two major categories: *active students* and *viewers*. The second levels are based on the different activities typically performed by the students in these two categories. For the "active students" we categorise them based on their final result. For the "viewers", we further divide the category based on their engagement quotient, i.e., how much of the course content they follow and whether they involve with the non-mandatory exercises in the course or not. Further, in this contribution we analyse the behaviour of the students in different categories to highlight the basic differences among them.

Keywords

Student categorisation, Student achievement, Massive open online courses, Student engagement

1. INTRODUCTION

The global wave of free, large and virtual courses attracts an incredibly diverse student population. With this diversity comes a huge variety of online behaviours. For data scientists it is a challenge to find categories that are suitable for sampling the whole population. It is also important to keep the categorisation scalable and robust.

To the best of our knowledge, there exist only a few categorisation schemes, mostly based on what emerges as a pattern of behaviour from MOOC students. These categories are based on the students' motivation [10] or engagement patterns [6, 7, 9, 4, 3, 5] or demographics [2, 1].

Based on student motivation (their "stated intent") of the students, [10] categorised the students, No-shows, Observers, Casual Learners and Completers. Where No-shows only register, Observers want to know about how a MOOC looks like, Casual Learners want to learn a few things only, and Completers want to earn a finishing certificate.

There are many categorisation schemes depending on engagement patterns. [6] categorised students in Completing, Auditing, Disengaging and Sampling students based on their activities which range from watching majority of lectures and submitting all the assignments (Completing) to watching only one or two lectures and no assignment submissions (Sampling). In a connectivist MOOC setting, [7] categorised students into Active (students who adapt well to the connectivest pedagogy), Passive (frustrated ones) and Lurkers (who actively follow the course but do not interact with anyone). Phil Hill first categorised MOOC students into Lurkers (ones who only enrol or sample the course), Active (fully engaged with the course material, quizzes and forums), Passive (only consume the content, did not participate in forums) and Drop-ins (consumed only a part of the course as an Active student) [5]. Later he revised his categories and divided the Lurkers into No-shows and Observers [3, 4].

Petty and Farinde [9] used the engagement categories from [8] to categorise students in an online mathematics course. These categories, based on the students' engagement patterns into critical thinking, were Clarification, Assessment, Inference, and Strategies.

The other dimension used to categorise students is to look at the demographics. For an electrical engineering course [2] categorised students based on their country of origin, education qualifications and backgrounds. Looking at the demographics of University of Pennsylvania's Open Learning Initiative [1] also categorised MOOC students based on their country of origin and educational background as [2] did. However, [1] added a few more categories based on gender, age and employment status of the MOOC students.

One common feature about these categorisation schemes is that they all consider only one of the dimensions of student behaviour, for example, engagement with the course content or forums or demographics or motivation. In this contribution, we present a novel categorisation scheme that considers both the engagement and the achievement of MOOC students. We further report on the different patterns shown by

the students from different categories. Moreover, the categories like Completing [6] and Active [4] are more than just engagement patterns; they also represent a mixed population of students with some achievement "flag". Therefore, we propose to further divide this category into subcategories based on the students' achievement.

2. RESEARCH QUESTIONS

In this study, we ask two main research questions:
Question 1: How can we categorise the MOOC students into categories that reflect both their achievement and engagement?
Question 2: What are the basic differences in the online behaviour of the students representing populations from different categories? More specifically, we are interested in finding the different ways to succeed in a MOOC which leads us to the following research questions. **Question 2.1** How does the engagement with the course content relate to the achievement? **Question 2.2** How does the timing of engagement i.e., the engagement with the course structure relate to the achievement? **Question 2.3** How does the effort during graded assignments relate with the achievement?

3. COURSE DETAILS

For this analysis we chose four courses from Coursera. The courses were basic JAVA and C++ both at the fundamental levels and as an introduction to object oriented programming. The courses were in French and were developed at École Polytechnique Fédérale de Lausanne, Switzerland. All the courses were basic level programming courses. All the courses had 7 weeks of lecture material. All the courses had programming assignments to grade the students. Also they had additional non-graded quizzes for practice. All the courses had the last deadline in the 11^{th} week from the beginning of the course. They also had soft deadline for the programming assignments after which the effective submission score reduced to 50 % of the actual score. All the courses were open after the final deadline as well.

4. CATEGORIES

We propose a hierarchical categorisation scheme. The first reason for having a few second levels in the scheme is to be able to include the achievement of MOOC students in the analysis of online behavioural patterns. The existing categorisation schemes lack on this front. They put the completion of the course as the only criterion for having a category, which oversimplifies the different levels of achievement. Having more levels for the students' achievement enables us to identify the different trends to succeed in a MOOC.

We have two first level categories: active students and viewers (based on whether the student participated in the grades assignments or not). Active students are subcategorised based on their achievement levels and viewers are subcategorised based on their further engagement with the course content. The motivation for subcategorising viewers was to have equally distributed categories so that none of the categories have a vast majority of the student population. This improves the generalisation of the categorisation schemes beyond the courses we chose to establish the categories.

We divide the whole student population in two major categories. First, those students who actively participate in the course, i.e., they take part in the assessment processes. We simply call these students "Active students". The active students get an achievement label at the end of the course. Second, those students who just watch the videos from the course (irrespective of the number of videos they watch). We call these students "Viewers". The viewers do not get any achievement label at the end of the course.

We further divide the active students based on their achievement labels that they get at the end of the course. Active students can either be "failed", "normal", or "distinction". The levels of "normal and distinction students may vary from on course to another, but for the courses we chose the criteria is the same for differentiation of these two subcategories of active students. Moreover, all the data for the active students is collected between the start week of the course and the last week of the assignment submission deadline.

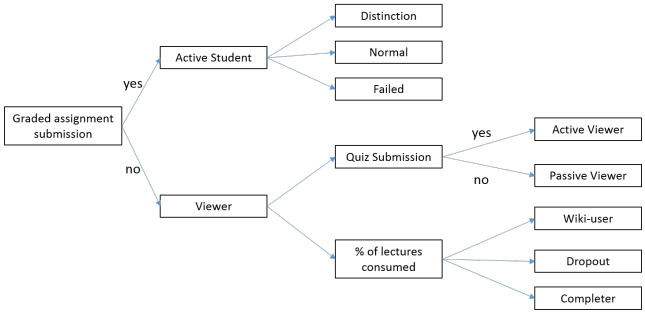


Figure 1: Hierarchy used in the present categorisation scheme.

The viewers, are further divided based on two factors. First, the amount of videos they watch; and second, whether they assess their learning by the means of the non-mandatory quizzes (in-video quizzes or regular non-graded quizzes) or not. Using the first factor, we divide the students into: 1. "wiki viewers" (if a student watches less than 10% of the videos). 2. "dropouts" (if a student watches between 10% and 70% of the videos). 3. "completers" (if a student watches more than 70% of the videos).

Using the second factor, we divide the the student into "Active Viewers" and "Passive Viewers". Since the courses were open even after the last assignment deadline, we consider the data till date of data export from Coursera (20^{th} week) for analysing the behaviour of the viewers.

5. VARIABLES

We used the following variables to analyse the behaviour of the students in different categories:

5.1 Active students

For analysing the differences in the activities among different achievement levels of Active students we defined the **First submission score:** the average score of the first attempt of all the programming assignments, as a proportion of maximum attainable score for each assignment. **First action week:** the first week of any kind of activity after registering for the course, once the course had started. **Activity span:** the difference in weeks between the first activity (as described in the previous item) and the last activity.

Progress within programming assignments: the difference between the two consecutive submissions for the same assignment, as a proportion of maximum attainable score for each assignment. Average number of attempts for each programming assignment. Proportion of videos watched **Delay in watching the lectures:** the time difference in weeks, between the time when the video was released online and the time the students watched it for the first time. Number of forum Views. **Procrastination index:** the ratio of the time difference between the submission time and the hard deadline and time difference between assignment being posted online and the hard deadline.

5.2 Viewers

For analysing the differences across the viewers' subcategories, we use only four of the above mentioned variables: first action week, delay in watching the lectures, activity span and the number of forum views.

6. RESULTS

In this section, we describe the differences between the different levels of active subcategories and viewer subcategories.

6.1 Active students

Concerning the lecture activities, the number of lectures watched by the failed students is significantly lower than the students having normal passing grades or the students with distinction $F[(2, 9914) = 741.95, p < .001]$. The lecture delay (overall and across the 7 weeks of lectures) decreases significantly as we move from distinction to normal to failed students $[F(2, 9914) = 91.43, p < .001]$.

Concerning assignment submissions, we see many differences across the three achievement levels. The first submission score decreases significantly as we move from distinction to normal to failed students $[F(2, 9914) = 210.65, p < .001]$. Number of attempts decreases significantly as we move from failed to distinction to normal students [F(2,2,9914)=222.86, $p < .001$]. The average improvement in two consecutive submissions for the same assignment is significantly higher for the students with distinction than the students with normal and failed levels $[F(2, 9914) = 101.58, p < .001]$. Moreover, the average procrastination index for the students with distinction level is significantly lower than the students from other two subcategories $[F(2, 2, 9914) = 343.83, p < .001]$.

The probability of achieving a higher grade decreases as the first action week approaches the 11^{th} week $[\chi^2(N = 9917) = 201.73, p < .001]$. The activity span for failed students is significantly smaller than passed students (normal and distinction) the course $[F(2, 2, 9914) = 972.68, p < .001]$. If we look at the forum views, the average number of forum views decreases significantly as we move from distinction to normal to failed students $[F(2, 2, 9914) = 135.42, p < .001]$.

6.2 Viewers

The viewer subcategories are based on two factors; first, how much video content they watch and second, whether they participate in non-mandatory quizzes or not. Here we present the results of the different activities for the viewer subcategories. The wiki-users tend to be passive viewers and completers tend to be active users $[\chi^2(N = 35, 193) = 4322.85, p < .001]$.

We observed an interaction effect of the two viewer subcategories on the first action week $[F(2, 35187) = 95.60, p < .001]$. For passive wiki-users and completers the first action week is significantly higher than the active wiki-users and completers. However, we see the opposite trend for the active and passive dropout viewers.

There were two single effects for the two viewer sub-categories on the activity spans. The activity span is more for the active viewers than the passive viewers $[F(1, 35191) = 1484.3, p < .001]$. Also, the activity span increases significantly as we more from wiki-users to dropouts to completers $[F(2, 35190) = 1919.63, p < .001]$.

There was an interaction effect of the two viewer sub-categories on the lecture delays $[F(2, 35187) = 67.50, p < .001]$. For passive wiki-users and completers the first action week is significantly higher than the active wiki-users and completers. However, we see the opposite trend for the active and passive dropout viewers.

7. DISCUSSION

We show that there are clear differences across the subcategories of active students and viewers. Active students are further subdivided into failed, normal and distinction categories. In section 3.1, we can see that the three categories are very different in terms of lecture, assignment, forum activities as well as their timing of these activities. What emerges from the results that the final achievement label that the active students get depends on a number of factors: 1) initial score, 2) engagement with the course content and forums, 3) efforts in assignment submissions and 4) timing of the activities. The variables we chose to differentiate among the achievement subcategories cover all these factors.

The distinction students get higher scores in their first submissions for the graded assignments than the normal and failed students, they improve more than the other two categories within two consecutive submissions for the same assignments and hence they reach the maximum attainable grade in fewer attempts. This reflects the effect of the initial score and efforts on the achievement level (**Question 2.2**). On the other hand, in spite of having similar improvements to the failed students the normal students get a better achievement level because of submitting more number of times. This shows the relationship between efforts and achievement (**Question 2.3**). Moreover, the distinction students have lower procrastination index for all the assignments than the other two categories. This reflects the relation between engagement with the structure (**Question 2.3**) and the achievement level.

The students who pass the course (distinction and normal) watch more videos than the students who fail. This simply reflects the fact that the students who pass the course engage more with the course content than those who fail the course, and establishes a relation between the engagement with the course content and achievement (**Question 2.1**). More interesting fact is that there is almost no difference between the distinction and normal students in terms of en-

gagement with the course content, however, there is a big difference in the delays that the students display in watching the video lectures. The distinction students have a smaller delay, especially in weeks 2 to 6, than the normal students. This shows the that there is a effect of engagement with the course structure (**Question 2.2**) on the achievement level.

Furthermore, the distinction students visit forums more often than the students from other two categories and the passed students (distinction and normal) have longer activity span than the failed students. It also reflects the effect of engagement on the achievement level (**Question 2.1**).

We see some peculiar behavioural patterns for viewers. One clear relation we see is between the engagement level and the activity span of the viewers. The passive users have smaller activity span than the active users. This simply translates to the fact that the people who assess their knowledge in some manner they tend to engage longer with the course content. We observed this fact for all the viewers.

The wiki-users have a very short activity span. This could be explained in two ways: either they started the course very late and realised that they can not pass the course and hence they left; or, they look for very specific content, look at a few videos for the required content and leave the course. The second behaviour is very similar to a Wikipedia user who looks for a very specific piece of information, obtains it and leaves the website. This was the main reason we called this category wiki-users. The passive wiki users start the course very late (only earlier than the passive completers), have an activity span of less than a week, i.e., they visit the course for some very specific content, then leave the course, this behaviour is closer to what we called a wiki-user's behaviour.

The completers display very interesting patterns, viewers in this category watch more than 70% of the video lectures. The difference in the activity spans of passive and active completers is about 4 weeks, this can be explained by the fact that the passive completers are only interested in the content and not in any kind of self assessment, hence they go through the whole content at a very high pace.

There are some overlaps between the categories we propose and the categories proposed by other researchers. For example, the wiki-users are similar to the sampling in [6] and observers in [4, 3]. Similarly, dropouts are a midway (or a mixed population of) category to disengaging in [6] and drop-ins in [4]. The passive viewers are similar to auditing and passive in [6] and [3] respectively. The completing category in similar to active students and completers in viewer population are similar to auditing [6]. However, the main motivation of putting these two in different categories was to capture there different activities which are clearly driven by different motivations, for the active students the main motivation is to get a certificate and for the completers in viewer population just want to watch the videos as a source of knowledge but do not want a completion certificate.

8. CONCLUSIONS

We presented a new MOOC student categorisation scheme. Its basic idea is to have a hierarchy to categorise MOOC students. We used both engagement and achievement to achieve this goal. First, we categorise students into two broad categories active students and viewers. Active students are those who submit graded assignments and viewers do not take part in this process. Further, we divide active students into normal, distinction and failed students, based on their grades; and we divide viewers into active and passive viewers (whether they attempt quizzes or not) and into wiki-users, dropouts and completers (based on how many video lectures they consume).

Throughout our analysis, we highlight the basic activity differences between subcategories of active students and viewers, proposing a few novel variables, like delay in watching lectures and procrastination index. We identify the different paths of success for the active students and different styles for the viewers. One clear difference between the proposed categories and existing categories is that in all the existing categories there is one category that contains a majority of the student population; whereas in the categories we propose, there is no such category.

The present categorisation scheme might have long term implications. First, for initiating a feedback system for those who dropout midway out of a course, we need a benchmark behaviour to compare against. The online behaviour of the students who passed and/or the completers in the viewer categories can be used in such cases. From the differences among different subcategories we report, it is clear that the different behaviour tend to start emerging as early as from the second week. This can be used to proactively help those students who are lagging behind in their engagement with the course content and course structure.

9. REFERENCES

[1] G. Christensen, A. Steinmetz, B. Alcorn, A. Bennett, D. Woods, and E. J. Emanuel. The mooc phenomenon: who takes massive open online courses and why? *SSRN 2350964*, 2013.

[2] J. DeBoer, G. S. Stump, D. Seaton, and L. Breslow. Diversity in mooc students? backgrounds and behaviors in relationship to performance in 6.002 x. In *Sixth Learning International Networks Consortium Conference*, 2013.

[3] P. Hill. Emerging student patterns in moocs: A graphical view, 2013.

[4] P. Hill. Emerging student patterns in moocs: A (revised) graphical view, 2013.

[5] P. Hill. The four student archetypes emerging in moocs. *E-Literate. March*, 10, 2013.

[6] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Third international conference on learning analytics and knowledge*. ACM, 2013.

[7] C. Milligan, A. Littlejohn, and A. Margaryan. Patterns of engagement in connectivist moocs. *MERLOT Journal of Online Learning and Teaching*, 9(2), 2013.

[8] C. Perkins and E. Murphy. Identifying and measuring individual engagement in critical thinking in online discussions: An exploratory case study. *Educational Technology & Society*, 9(1), 2006.

[9] T. Petty and A. Farinde. Investigating student engagement in an online mathematics course through windows into teaching and learning. *Journal of Online Learning and Teaching*, 9(2), 2013.

[10] J. Wilkowski, A. Deutsch, and D. M. Russell. Student skill and goal achievement in the mapping with google mooc. In *First ACM conference on Learning@ scale conference*. ACM, 2014.

Analyzing student inquiry data using process discovery and sequence classification

Bruno Emond
National Research Council Canada
bruno.emond@nrc.gc.ca

Scott Buffett
National Research Council Canada
scott.buffett@nrc.gc.ca

ABSTRACT

This paper reports on results of applying process discovery mining and sequence classification mining techniques to a data set of semi-structured learning activities. The main research objective is to advance educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. As an example of our current research efforts, we applied temporal data mining analysis techniques to a PSLC DataShop data set [17, 18, 19, 20]. First, we show that process mining techniques allow for discovery of learning processes from student behaviours. Second, sequential pattern mining is used to classify students according to skill. Our results show that considering sequences of activities as opposed to single events improved classification by up to 230%.

1. INTRODUCTION

The Learning Performance Support Systems program (LPSS) at the National Research Council Canada aims at delivering a personal learning environment (LPSS.me), software algorithms, and prototypes to enable Canada's training and development sector to offer learning solutions to industry partners that will address their immediate and long-term skills challenges. The main elements of the personal learning environment include a common platform architecture, a personal learning assistant, a personal cloud, learning resources repository network, personal learning records, and analytics to discover and assess competencies. The program is at an early stage of development.

One of the main thrusts within this research program seeks to advance and apply educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. Our initial position points towards a complementary use of latent knowledge estimation and performance prediction methods [3], and temporal data mining methods. A main research trend in educational data mining consists of analyzing students' performance within intelligent tutoring systems, focusing on the correctness of previous questions or the number of hints and attempts students needed in order to predict their future performance [6]. Predictive mathematical models resulting from this analysis characterize, through parameter values, some information contained in the sequence of actions leading to student performances, but do not represent explicitly those sequences. Over the years there has been a growing interest to examine explicitly learning sequences as a complementary approach. Process and sequence mining have been applied for the analysis of content sequencing and curriculum sequencing [5, 15], group behaviour sequences in collaborative software development tasks [16], problem solving behaviours over a shared tabletop [14], as well as self-regulated learning and meta-cognition [7].

The remainder of this paper consists of a short presentation of temporal data mining, followed by process mining and sequence mining analyses of a semi-structured inquiry learning activity data set [17, 18, 19] obtained from the Pittsburgh Centre for Science and Learning DataShop [8]. We show that process mining techniques allow for the discovery of learning processes, and that sequential pattern mining can used to identify the level of skill exhibited by each student.

2. TEMPORAL DATA MINING

Temporal data mining refers to the extraction of information and knowledge from potentially large collections of temporal or sequential data [12]. According to Laxman and Sastry [9], sequential data refers to any type of data where data points are explicitly ordered, either by time stamps or some other sequencing mechanism. This includes data such as moves in a chess game or commands entered by a computer user, but also other forms of data that are not explicitly time-stamped but are still otherwise ordered, such as text or protein sequences.

Temporal data is often divided into two categories: sequences that consist of continuous, real-valued data points taken at regular intervals, which are referred to as *time series data*, and sequences that may be represented by compositions of nominal symbols from a particular alphabet, which are referred to as *temporal sequences* [2]. As the field of time series analysis has a long history with many established techniques, the more recent field of temporal data mining instead focuses on information extraction from temporal sequences.

Given a set of temporal sequences, the general tasks of tem-

poral data mining consist of 1) prediction, 2) classification, 3) clustering, 4) search and retrieval, and 5) pattern discovery. These tasks can be accomplished using a number of established techniques in the area. A few of the more prevalent techniques include: A) *Sequential pattern mining*: The goal of sequential pattern mining [1] is to identify highly frequent sequences that appear within a database of ordered items or events; B) *Sequence classification*: Sequence classification [11] attempts to assign a candidate sequence to one of possibly several classes of existing sequences, typically according to either similarity or common features such as frequent sub-sequences; C) *Episode mining*: Frequent episodes [13] are sets of partially ordered events that are found to occur close together frequently and consistent with the specified partial order; and D) *Process mining*: Process mining refers to the extraction of process-related information from event logs [21]. Process mining algorithms are used to build a model of the business process by representing the different ways cases in the process can be executed. However, there are some key differences between business processes and learn flows [4].

3. TEMPORAL EDM ANALYSIS

To demonstrate the potential of temporal data mining in the analysis of educational data, we conducted a study utilizing process mining and sequential pattern mining to discover learning processes and to identify the level of student skill using a data set [17, 18, 19] taken from the Pittsburgh Science of Learning Center DataShop [8]. This data set contains data on 148 middle school students performing activities logged while working within a micro-world, where students engage in "scientific inquiry" to study liquid phase change. Here, the students form hypotheses and conduct experiments as they investigate whether container size, heat level, substance amount, and cover status affected the boiling/freezing point of water, or the time it took to freeze/boil. All students' fine-grained actions were attributed a time stamp and recorded by the system. These actions included: interactions with the inquiry support widgets, interactions with the simulation including changing simulation variable values and running/pausing/resetting the simulation, and transitioning between inquiry tasks [18].

Given that we are mostly interested in the discovery of self-regulated learning, the fact that students had a moderate degree of freedom to choose their own procedures for conducting experiments, less than in purely exploratory learning environments though [19], was an interesting data set for studying sequences of student behaviours and how they correlate with student success.

3.1 Process Mining and Discovery

Process mining offers a set of techniques and tools to discover sequential patterns represented as workflows. The analysis in this section was performed using the *Inductive visual miner* [10]. We were interested to discover, from the log of students inquiry activities, similar process models to the one depicted in Figure 1. For this discovery analysis, we limited ourselves to the whole data set, and we did not try to distinguish between groups of students. The purpose was to explore and compare the actual processes that students followed to the expected process from the author of the learning environment given in Figure 1, rather than suggest alternative learning processes. The log file contained 29679 events for 147 students. The overall distribution of inquiry activities indicated that 58.1% were spent in analysis, 19.1% in experiment, 18.4% in hypothesis formation, and 4.4% in observation.

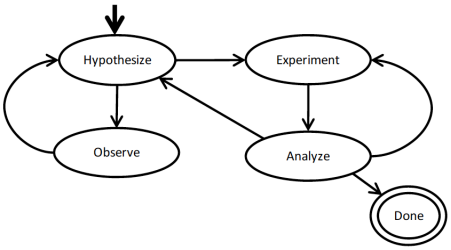


Figure 1: Intended learning paths during scientific inquiry.

As indicated in Figure 1, the intended learning process contains many possible loops while students progress in their scientific inquiry. Figure 2 and Figure 3 show respectively discovered process models from the transactions log using 100% of the events and sequences, and the top 70% most frequent events and sequences. From the visual comparison of the process model for 100% of the data (Figure 2), and the intended process of Figure 1, it is clear that there is a lot of variability in students transitioning between inquiry steps, given that the model is mostly disjunctive, with sequences resulting from loops. However, after leaving out the 30% most infrequent events and event sequences from the data, we discover a process model, Figure 3, that has some resemblance to the intended inquiry process, representing explicitly the sequence of hypothesize to experiment or analyze. Notice that the observation inquiry step is not part of the model because of the low frequency of its related events, which indicates a difference with the intended learning process, or more accurately, a tendency by the students to avoid the observation stage.

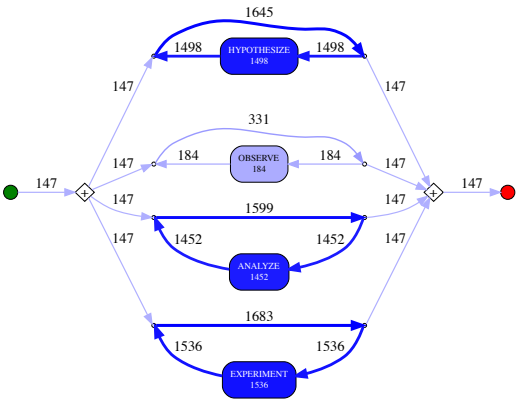


Figure 2: Process model using 100% of events and sequences (from top to bottom: hypothesize, observe, analyse, experiment).

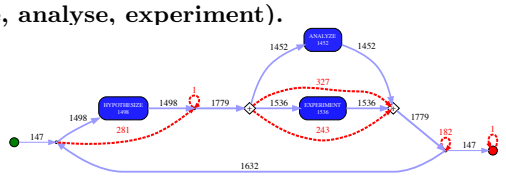


Figure 3: Process model using 70% of most frequent events and sequences (from left to right: hypothesize, analyse (top), experiment (bottom)).

Another element of interest was the sequence of problems students address during their inquiry. The overall distribution of student activities within those problems were relatively balanced with 30.7% in "container size", 24.9% in "amount of substance", 23.0% in "level of heat", and 21.4% in "cover status". Figure 4 shows a process model including 100% of events and event sequences. The process model clearly indicates a bias towards starting from the container size problem, followed by equivalent choices from the three other problems. This is likely a consequence of the the container size being the default value at the start of the inquiry session, which is a restriction on the student self-regulated learning processes.

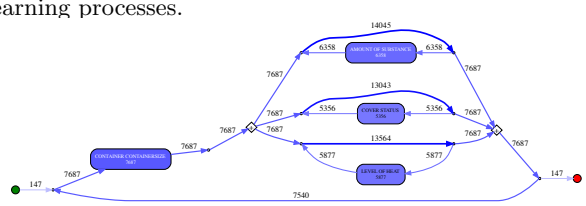


Figure 4: Process model of problems sequence using 100% of events and event sequences (from left to right: container size, amount of substance (top), cover status (middle), level of heat (bottom)).

Interestingly though, one would expect that the inquiry steps would be grouped (follow each other closely) within each problem. An inspection of a process model for an event classifier including the combination of both inquiry steps (hypothesize, observe, experiment, analyze) and problems (container size, amount of substance,level of heat, cover status) with 100% of events and sequences reveals only three groups of steps and not four as one would expect. In Figure 5, 1) the leftmost group is focused on inquiry steps applied to container size, and amount of substance, 2) the middle group to level of heat, amount of substance, and cover status, and 3) the rightmost group to cover status. This distribution of steps indicates that the four problems were not explored completely independently by the students, which manifest a strategy to explore concurrently the effect of different factors. However, this strategy might be different when comparing students with good and poor results and should be explored in a subsequent analysis.

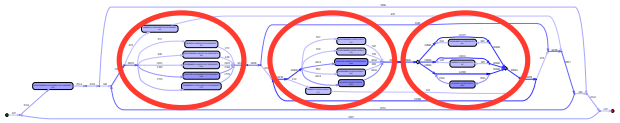


Figure 5: Three groups of problems and inquiry steps combination sequences.

3.2 Sequence classification

The second phase of our study was to explore the potential of sequential pattern mining in the identification of the level of skill exhibited by each student. Since sequences of student activity in the data set were not explicitly labelled as "skilled", "unskilled", etc., we considered two other metrics to measure skill exhibited: 1) number of times the student got an answer wrong, and 2) total time taken to complete the experiments. We used leave-one-out cross validation, applying our sequence classification learning algorithms on the training set and attempting to classify each test student as having either the high/low number of incorrect answers, or high/low time to complete, depending on the test.

Figure 6 shows the results of classifying students as "high number of incorrect steps". Success of the classifiers are measured by likelihood ratio (LR), which indicates how much more likely a positive example will be classified as positive than a negative example. The left-hand chart shows the success in classifying whether a student is in the bottom 50% in terms of number of incorrect answers, for varying maximum sequence size. Thus, a maximum sequence size of 1 represents the case where sequential relations are not considered, and only the presence/absence of certain actions are used for the classification. Observe that the LR is close to 1 in this case, meaning that we are no more likely to classify a positive case as positive or negative. The LR then increases steeply by 230% to 2.3 as sequences of size 2 are considered, before levelling off at about 1.75 for size 3 and greater. The right-hand chart then demonstrates how the classifier improves as we use sequences (max size 4) to classify students into the categories of worst 50%, 40%, 30%, 20% and 10%. Figure 7 depicts the results similarly for classifying students as "long time to complete". While not as dramatic, the positive effect of utilizing sequential information is demonstrated here as well.

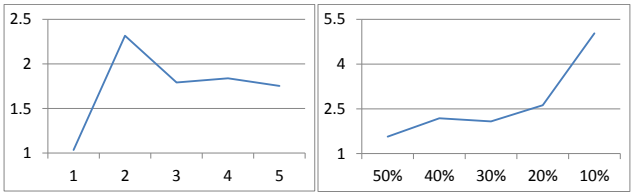


Figure 6: LR for classifying as "high number of incorrect steps".

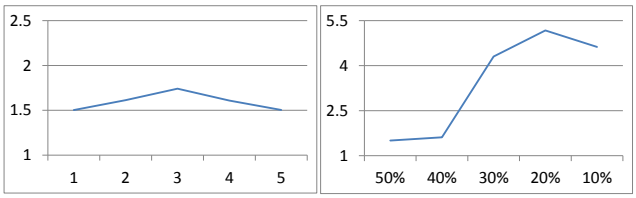


Figure 7: LR for classifying as "long time to complete".

4. CONCLUSION

One of the main thrusts within the Learning Performance Support Systems program (LPSS) at the National Research Council Canada seeks to advance and apply educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. The program is at an early stage of development and our initial position points towards a complementary use of latent knowledge estimation and performance prediction methods [3], and sequence mining methods. In order to support the validity of our argument that sequential data analytics holds great potential for the analysis of student knowledge and skill acquisition, we demonstrated the application of discovery process mining and sequence mining in classifying students according to success using a data set of semi-structured learning activities [17, 18, 19] taken from the Pittsburgh Science of Learning Center DataShop [8].

Using process mining tools we were able to discover in-

quiry learning patterns in relationships with inquiry learning steps, learning problems, and a combination of those. Our analysis showed some differences between the semi-structured process intended by the developers of the learning environment and the actual processes followed by the students. We also showed that process mining techniques allow for the discovery of learning processes, and that considering sequences of events as features we can improve classification by up to 230% over considering single, non-sequential events. Given the learning process patterns discovered in the initial analysis of the students inquiry activity log, the next process mining discovery analysis will be to compare the inquiry processes of students having low and high correct outcomes.

5. ACKNOWLEDGEMENT

We would like to thank the Pittsburgh Science of Learning Center for providing the data supporting this analysis. We used the 'Science Sim State Change January 2010' data set accessed via the PSLC DataShop [8]. We thank Ken Koedinger from Carnegie Mellon for his help in choosing this data set. This work is part of the National Research Council Canada program Learning and Performance Support Systems (LPSS), which addresses training, development and performance support in all industry sectors, including education, oil and gas, policing, military and medical devices.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th Int'l Conference on Data Engineering*, pages 3–14. IEEE, 1995.

[2] C. M Antunes and A. L. Oliveira. Temporal data mining: An overview. In *KDD workshop on temporal data mining*, pages 1–13, 2001.

[3] R. S. Baker and A. T. Corbett. Assessment of robust learning with educational data mining. *Research and Practice in Assessment*, 9:38–50, 2014.

[4] R. Bergenthum, J. Desel, A. Harrer, and S. Mauser. Modeling and mining of learnflows. In K. Jensen, S. Donatelli, and J. Kleijn, editors, *LNCS Transactions on Petri Nets and Other Models of Concurrency*. Springer, Springer-Verlag : Berlin Heidelberg, 2012.

[5] M. C. Desmarais and R. S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modelling and User-adapted Interaction*, 22:9–38, 2012.

[6] H. Duong, L. Zhu, Y. Wang, and N. T. Heffernan. A prediction model that uses the sequence of attempts and hints to better predict knowledge: "better to attempt the problem first, rather than ask for a hint". In *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013* [6], pages 316–317.

[7] J. S. Kinnebrew, K.M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5:190–219, 2009.

[8] K.R. Koedinger, R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. *A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining*. CRC Press, Boca Raton, FL, 2010.

[9] S. Laxman and P. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[10] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst. Process and deviation exploration with inductive visual miner. In *In Twelve International Conf. on Business Process Management, Accepted Demonstration 46*, Eindhoven, Netherlands, 2014.

[11] N. Lesh, M. J. Zaki, and M. Ogihara. Mining features for sequence classification. In *Proc. of the fifth ACM SIGKDD international conf. on Knowledge discovery and data mining*, pages 342–346. ACM, 1999.

[12] N. Mamoulis. Temporal data mining. In Ling Liu and M Tamer Özsu, editors, *Encyclopedia of Database Systems*. Springer, New York, NY, 2009.

[13] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences extended abstract. In *Proceedings the first Conference on Knowledge Discovery and Data Mining*, pages 210–215, 1995.

[14] R. Martinez, K. Yacef, J. Kay, A. Al-Qaraghuli, and A. Kharrufa. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Proc. of the Fourth Int'l Conference on Educational Data Mining*, Eindhoven, Netherlands, 2011.

[15] M. Pechenizkiy, N. Trcka, P. De Bra, and P Toledo. Currim: Curriculum mining. In *Proceedings of the 4th International Conference on Educational Data Mining*, pages 216–217, 2012.

[16] D. Perera, J. Kay, I. Koprinska, K. Yasef, and O. Zaiane. Clustering and sequential pattern mining to support team learning. *IEEE Transactions on Knowledge and Data Engineering*, 21:759–772, 2009.

[17] M. Sao Pedro, R. Baker, and J. Gobert. Improving construct validity yields better models of systematic inquiry, even with less information. In J. Masthoff, B. Mobasher, M. Desmarais, and R. Nkambou, editors, *Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization*, pages 249–260, Montreal, Canada, 2012.

[18] M. Sao Pedro, R. Baker, and J. Gobert. What different kinds of stratification can reveal about the generalizability of data-mined skill assessment models. In J. Masthoff, B. Mobasher, M. Desmarais, and R. Nkambou, editors, *Proceedings of the 3rd Conference on Learning Analytics and Knowledge*, Leuven, Belgium, 2013.

[19] M. Sao Pedro, R. Baker, J. Gobert, O. Montalvo, and A. Nakama. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, 23:1–39, 2013.

[20] M.A. Sao Pedro. *Real-time Assessment, Prediction, and Scaffolding of Middle School Students' Data Collection Skills within Physical Science Simulations*, 2013.

[21] W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.

Desirable Difficulty and Other Predictors of Effective Item Orderings

Steven Tang Hannah Gogel Elizabeth McBride Zachary A. Pardos
University of California, Berkeley
Tolman Hall
Berkeley, CA, USA
{steventang, hgogel, bethmcbr, pardos} @berkeley.edu

ABSTRACT

Online adaptive tutoring systems are increasingly being used in classrooms as a way to provide guided learning for students. Such tutors have the potential to provide tailored feedback based on specific student needs and misunderstandings. Bayesian knowledge tracing (BKT) is used to model student knowledge when knowledge is assumed to be changing throughout a single assessment period. The basic BKT model assumes that the chance a student transitions from "not knowing" to "knowing" after each item is the same, with each item in the tutor considered a learning opportunity. It could be the case, however, that learning is actually context sensitive; context in our analysis is the order in which the items were administered. In this paper, we use BKT models to find such context sensitive transition probabilities in a mathematics tutoring system and offer a methodology to test the significance of our model based findings. We employ cross validation techniques to find models where including item ordering context improves predictive capability compared to the base BKT models. We then use regression testing to try to find features that may predict the effectiveness of an item ordering.

Keywords

Item Ordering, Bayesian Knowledge Tracing, Item Difficulty

1. INTRODUCTION

Online adaptive tutors are increasingly being used in classrooms as supplements to traditional instruction. Some systems, such as the ASSISTments [4] platform used for middle school math subjects, provide scaffolding or hints to students upon request or when the student answers a question incorrectly. In this paper, we focus on employing the Bayesian knowledge tracing (BKT) model of student learning but with the hypothesis that learning could be *context sensitive*. In this case, the context is the order that items of a particular skill are administered in.

2. BACKGROUND

2.1 ASSISTments Data

The data set analyzed in this paper comes from use of the ASSISTments platform in AY 2012-2013. The data set is publicly available and is rich with information that has been mined by other research projects [7] [9]. In this paper, we focus on the *Skill Builder* sequences used in ASSISTments, where a problem set consists of items given in a random order, generated from a set of templates. Items generated from these templates are assumed to be answerable with knowledge of a single underlying knowledge component (KC). For example, one problem set might contain three item templates. Each template can be populated with a set of numbers to generate an item; thus many different items can be derived from a single template. The number of templates per problem set varies; in this paper, we look at problem sets with between 2 and 6 templates. The number of items delivered to the student depends on the student's performance; in the Skill Builder set, mastery is assumed to occur after three consecutive correct responses. Each template in a Skill Builder sequence has an associated method of assistance; it is either a *hint* template or a *scaffolding* template. Scaffolding templates are bundled with a set of simpler questions to guide the student through the ideas in the item, while hint templates have guiding statements available to assist the students (usually the final hint provides the exact answer to the item).

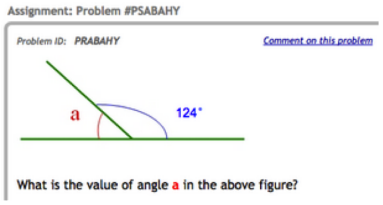


Figure 1: Example of an item in the ASSISTments database

3. METHODS AND ANALYSIS

3.1 Bayesian Knowledge Tracing

Bayesian knowledge tracing [3] assumes a binary representation of student knowledge. Figure 2 depicts a BKT model representation as a hidden Markov model (HMM). The basic BKT model is shown inside the dashed portion of the figure. O_1 through O_4 are binary indicators of correctness at opportunities 1 through 4. K_1 through K_4 represent the latent knowledge of the KC (assumed to be 0 or 1) at opportunities 1 through 4. In between each K_i and K_{i+1}, there is an arrow representing a probability of *transition*, or learning. Guess and slip parameters can be assumed to be equal among all items or can be item-specific [10].

3.2 Item Ordering Effects

The Skill Builder sequences in the ASSISTments platform pick from a set of templates at random to generate items for

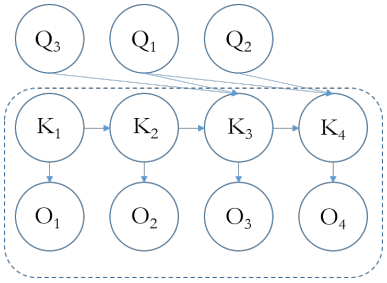


Figure 2: BKT Model. The dashed portion represents the basic BKT model, and the Q nodes represent the item order modification.

the student. However, it is our hypothesis that there may exist pedagogically more advantageous orderings of problems than the default random orders. Data mining and learning analytics techniques have been used to create process models and determine the most effective order of events for learners in online science education [8], as well as for finding patterns where students exhibited patterns of self-regulated learning [6]. Investigating the effects of item ordering can help both researchers and teachers, bridging the gap between educational theory and practice.

The BKT model could be extended to model a transition probability per particular item ordering. For example, one student might receive items from templates in the order of (3, 1, 2, ...) while another student might receive items from templates in the order of (1, 3, 2, ...). Over a number of such permutations, the BKT model could estimate a separate transition probability associated with items in the order (3, 1) as opposed to (1, 3). Figure 2 depicts how this new model might be formulated as an HMM, where items in the order of (3, 1, 2) are seen by the student. Note that the probability of knowledge at K_3 is influenced by seeing question 3 followed by question 1. Other students will be given items in different and random orders, allowing for all possible combinations of item order pairs to be analyzed. This model is drawn from work by Pardos and Heffernan [9]. We extend this work by finding significant improvements in predictive accuracy with the item order model by looking at the mean absolute errors produced by both the basic BKT and the item order model.

3.3 BKT model fitting

Among the Skill Builder response sets (SBs) from the 2012-2013 ASSISTments data set, we only looked at sets with more than 2000 student responses, more than 250 students, and between 2 and 6 (inclusive) templates. There were 112 Skill Builders that met these criteria, with 130,496 student response streams and 606,948 responses. Two BKT models, estimated using the XBKT code base, were fit to each of the 112 SBs. The first model was standard BKT (baseline), where every item was assumed to have the same transition probability. In our standard BKT model, every template type was allowed to have its own guess and slip parameters. The second model allowed for both different guess and slips per template and different transition probabilities based on the previous two items administered. We enabled different guess and slips per template for our baseline model so that any difference between models would be attributed to the different item order learning transitions. Additionally, we modeled a transition probability for each template specifically when that template was the first item administered in the sequence.

3.4 CV prediction to identify item orders of interest

To obtain statistical confidence in the generalization of a certain item ordering to unobserved students, we performed 5-fold cross validation (CV) on the data. This process starts by fitting both base and item order BKT models on a randomly selected 80% of student response data, and then using the trained models to predict student responses in the held out 20%, called the test set.

By comparing the predicted responses to the actual responses, Mean Absolute Errors (MAE) were obtained for both the base and the item order models. The error rates were then compared using a paired t-test for each possible item order. Out of the 1789 possible item orders among all Skill Builder problem sets, 605 item orders were found to have statistically significant error differences between the two predictive models at the .05 level. Among the 605 item orders, 157 had their responses predicted better by the base BKT model (by an average rate of .0138), while the remaining 448 item orders had their responses predicted better when using the item order model (by an average rate of .0173). It is important to note that the item orders in this section include ordering situations where the same template is administered twice in a row. The result that a portion of the item orders had better response prediction when using the base BKT model is not surprising, considering that each addition of a single new template to an SB increases the number of potential item orders dramatically. Thus, as the number of templates increases, the number of responses per item order decreases, resulting in less data per parameter for the model to learn from. The occurrence of 448 item orders whose responses were better predicted by the item order model suggests that the item order model could be able to uncover effective (or ineffective) item orderings.

Figure 3 shows the distribution of learn rates from both the basic and the item order BKT models. In the basic BKT model, a learn rate represents the rate at which a student is expected to learn (if they did not already know it) the latent knowledge component after seeing any item. In the item order BKT model, learn rates are modeled per item order pair, thus representing the rate a student is expected to learn a knowledge component after seeing a particular order of two items. The combination of the item order model with the cross validation approach provides a procedure that can determine when the item order model provides more accurate predictions compared to the base BKT model. Such a procedure can reveal when an item ordering might be considered effective or ineffective.

3.5 Regression analysis

Regression analyses (212,858 student responses) were run on the 448 item orders found to be significantly better fitting from the cross validation approach in order to find predictors of the item order learn rates. For the regression analyses,

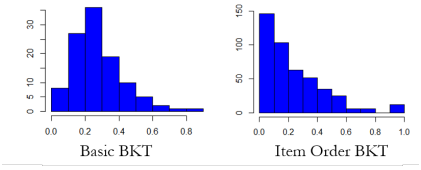


Figure 3: Distribution of learn rates

we extracted template level features from both templates in an item order. Features included are: average time to first response (milliseconds), percent correct on first problem attempt, average number of attempts, problem type (text response or radio button/multiple choice), difference in time to first response between Template A and Template B (where Template A is the first item in an ordering), difference in percent correct between Template A and Template B, whether Template A offered hints or scaffolding as assistance, and the *individual* learn rates for Templates A and B.

In our first model, stepwise regression was used regressing item order learn rate on these features (R^2=.17, F = 46.19, p<.01). The only features that were found to be significant at the .05 level were the learn rate of Template A, which had a negative effect on item order learn rate for the pair (β = -0.13, p = .01), and the learn rate of Template B, which had a positive effect (β = .502, p < .01) in the model.

Our second model only included item orderings where Template A was a scaffolding problem (R^2=.37, F = 11.47, p < .01). All of the features from the first model were included except for problem type due to lack of variation. Features unique to scaffolding problems were added as potential predictors: problem type of the associated sub-questions and percentage of scaffolding problems (including sub-questions) answered correctly. Average attempts on Template A (β = .93, p < .01) and the learn rate for Template B (β = .58, p < .01) had a positive effect on the item order learn rate. When the scaffolding for Template A consisted of text responses, the learn rate of the ordering decreased (β = -.13, p < .01).

The third model was fit using only orders where Template A was a hint item (R^2=.22, F = 20.94, p < .01). Hint features included percentage of students who went through all the hints on Template A and average amount of template hints seen. Average number of attempts on Template A (β = .27, p < .01), average milliseconds to first response on Template A (β = < .01, p = .03), percentage of students who accessed all of the hints on Template A (β = .71, p < .01), learn rate of Template A (β = -0.16, p <.01), and the learn rate for Template B (β = .43, p < .01) were significant predictors.

Regression analyses were also conducted to look for feature predictors of individual template learn rates for the 321 individual templates included in these 448 orderings. Percent correct on the template (β = .31, SE = .1, p < .01) and the item requiring a text response (β = .14, SE = .04, p < .01) were significant predictors (R^2=.06, F = 10.84, p < .01).

The primary unexpected result from the regression findings is that a *lower* learn rate of Template A predicts a higher learn rate for the ordering. It is important to note that this effect may be due to constraints in our current model. The individual learn rate of Template A is calculated when Template A occurs as the first item in a problem set presented to a student. That Template A is also included as part of an item ordering pair made up of the first and second items in the administered problem set. If the learn parameter for Template A is high, the knowledge component is already known (and has already been learned) by the time we consider the learn rate for the ordering including Template A. However, this phenomenon does not occur for template B of the item ordering, as Template B would not be the first template seen by the student in this case. In order to alleviate the discrepancy between the correlations, single template learn rates should be calculated from all template occurrences throughout administration in future work.

3.6 Desirable difficulty

In previous proof-of-concept work [11], a qualitative analysis was performed to examine what might make certain item orderings more effective than other item orderings. One feature of item pairs that became obvious was that not all items had exactly the same level of difficulty. In addition, some effective orderings contain a harder item first whereas other effective orderings contain an easier item first. One potential hypothesis that can help explain this difference in item ordering and difficulty is that of "desirable difficulties". In a series of studies, Bjork and colleagues determined that some challenges to performance during learning activities may actually contribute to greater learning [1] [2] [5]. By introducing "desirable difficulties" that help learners engage in the active processing of information, learning tasks that may be perceived as challenging or inefficient may prove more beneficial in the long run than those completed with high fluency.

In the case of item orderings where the first problem is more difficult than the second, the first (more difficult) problem may introduce a desirable difficulty, leading the student to learn more than they would with an easier problem. This learning then carries over into the second problem in the pair, thus leading to a higher overall rate of learning. This hypothesis works towards explaining our finding that a lower learn rate of the first template predicts a higher learn rate for an item ordering. When the first problem is easier than the second, this might be an instance where the material is better learned through a gentler or simpler introduction, as perhaps the second problem might be more difficult than is "desirable". In this case, a student would not properly learn from the more difficult problem unless it were preceded by an easier problem that would serve as a scaffold.

Using data from the BKT model to examine this hypothesis, we looked at how the difference between prior knowledge (at the start of an SB) and the percent correct on a template (as a proxy for template difficulty) compared to the probability of learning using regression. Finding *no difference* between a student's prior knowledge and the percent correct for a given template might show when an item has an "appropriate" difficulty. In this case, the difficulty of the item closely matches the prior knowledge of the student. Pedagogically, for an item to help the student learn, the difference between the student's prior knowledge and the item difficulty should be negative; in other words, the difficulty of the item should

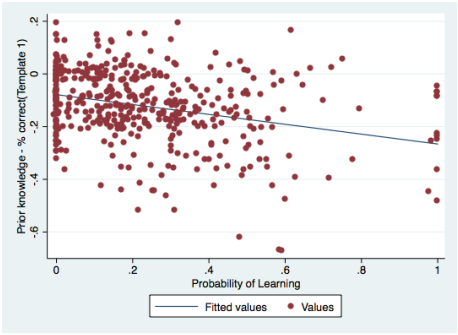


Figure 4: Scatterplot using template A data

be above the level of the student's prior knowledge to promote learning.

Regressing the difference between prior knowledge and item difficulty (percent correct) on the probability of learning showed statistical significance at the 0.01 level. This statistical significance held when using the difference between prior knowledge at the beginning of an SB and the percent correct on the first item in a pair (Template A), as well as the difference between prior knowledge and the percent correct for the second item in the pair (Template B). Using the percent correct for Template A to find the difference between the student's prior knowledge and the item difficulty had a correlation of -0.3039 with the probability of learning, while using Template B had a -0.2146 correlation with the probability of learning. These correlations are both relatively high, showing enough relationship between the variables to warrant further exploration in this area.

Similar to the correlations, the regressions were also run using percent correct from Template A and from Template B in the difference between prior knowledge and item difficulty. For Template A the coefficient for regressing the difference between prior knowledge and item difficulty (percent correct) on the probability of learning was -0.187 (R^2 =0.09, F=45.39); using template B, the coefficient was -0.120 (R^2= 0.046, F=21.53). The negative correlations, as well as negative coefficients in each of the regressions, show that the more negative the difference between prior knowledge and item difficulty becomes (the larger the difference between these two variables in the right direction for a "desirable difficulty"), the greater the probability of learning becomes. A scatterplot showing the relationship between these variables can be seen in Figure 4.

4. LIMITATIONS AND FUTURE WORK

The findings from this paper suggest that the item order BKT model combined with the use of a cross-validation technique show promise in uncovering learning mechanisms not apparent when just the base BKT model is used. The cross-validation approach confirmed that some item order models had better predictive capabilities compared to the base BKT models. Thus, statistically reliable suggestions can be made about item order delivery, and more research into item ordering is warranted, especially using such a cross-validation approach.

The results from the regression were somewhat surprising, where a lower individual learn rate from the first template in an ordering predicted a higher overall learn rate for the ordering. We hypothesize that this could be due to a constraint in our item order model, where individual learn rates of templates were modeled using only instances of that item when it appeared as the first item in a sequence. This hypothesis can be investigated in future research using a modified item order model.

5. REFERENCES

[1] M. Anderson, J. Neely, E. Bjork, and R. Bjork. *Memory, Chapter 6: Interference and inhibition in memory retrieval.* Academic Press, 1996.

[2] R. W. Christina and R. A. Bjork. *In D. Druckman and R. A. Bjork (Eds.), In the mind's eye: Enhancing human performance: Optimizing long-term retention and transfer.* Washington, DC: National Academy Press, 1991.

[3] A. T. Corbett and J. R. Anderson. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *The Journal of User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[4] M. Feng, N. Heffernan, and K. R. Koedinger. Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *The Journal of User Modeling and User-Adapted Interaction*, 19:243–266.

[5] V. Halamish and R. Bjork. When does testing enhance retention? A distribution-based interpretation of retrieval as a memory modifier. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37.

[6] K. L. J.S. Kinnebrew and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM-Journal of Educational Data Mining*, 4:190–219, 2013.

[7] J. Ocumpaugh, R. Baker, G. S., N. Heffernan, and C. Heffernan. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*, 45(3):487–501, 2014.

[8] L. M. P. Reimann and M. Bannert. e-Research and learning theory: What do sequence and process mining methods contribute? *British Journal of Educational Technology*, 45:528–540, 2014.

[9] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. In *Proc. of the 2nd International Conference on Educational Data Mining*, 2009.

[10] Z. A. Pardos and N. T. Heffernan. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP)*, 2010.

[11] S. Tang, E. McBride, H. Gogel, and Z. A. Pardos. Item ordering effects with qualitative explanations using online adaptive tutoring data. In *Proceedings of Works-in-progress at the second ACM conference on Learning@ scale*, 2015.

Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities

Ran Liu
Human-Computer Interaction Institute
Carnegie Mellon University
ranliu@cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
koedinger@cmu.edu

ABSTRACT
A growing body of research suggests that accounting for student-specific variability in educational data can improve modeling accuracy and may have implications for individualizing instruction. The Additive Factors Model (AFM), a logistic regression model used to fit educational data and discover/refine skill models of learning, contains a parameter that individualizes for overall student ability but not for student learning rate. Here, we show that adding a per-student learning rate parameter to AFM overall does not improve predictive accuracy. In contrast, classifying students into three "learning rate" groups using residual error patterns, and adding a per-group learning rate parameter to AFM, substantially and consistently improves predictive accuracy across 8 datasets spanning the domains of Geometry, Algebra, English grammar, and Statistics. In a subset of datasets for which there are pre- and post-test data, we observe a systematic relationship between learning rate group and pre-to-post-test gains. This suggests there is both predictive power and external validity in modeling these distinct learning rate groups.

Keywords
Student learning rate, learning curves, Additive Factors Model

1. INTRODUCTION
A growing body of research suggests that accounting for student-specific variability in statistical models of educational data can yield prediction improvements and may potentially inform instruction. The majority of work investigating the effects of student-specific parameters [6, 10, 11, 15] has been done in the context of a class of models called Bayesian Knowledge Tracing (BKT), a special case of using Hidden Markov Models to model student knowledge as a latent variable.

Logistic regression is another popular method for modeling educational data. The Additive Factors Model (AFM) [4] is one instantiation of logistic regression that was developed with the primary intention of evaluating, discovering, and refining *knowledge component (KC) models* (also referred to as Q-matrices). In contrast to *statistical models* of educational data, KC models define the knowledge components (e.g., skills, concepts, facts) on which estimates of students' knowledge are based. AFM has parameters modeling KC difficulty, KC learning rate, and individual student ability, but it does not have a parameter for individual student *learning rate*.

Recent work extending BKT models [15] suggests that better predictive accuracy is achieved by adding parameters that accommodate different learning rates for different students. Here, we investigate two different extensions of AFM that model student learning rate variability. The first model (AFM+StudRate) adds a per-student learning rate parameter to AFM, dramatically increasing the number of parameters in the model. We find some evidence that this model overfits the training data. For the second model (AFM+GroupRate), we introduce a method of classifying students into learning rate groups. We then add a per-group, rather than per-student, learning rate parameter to AFM and show that this model significantly outperforms regular AFM in predictive accuracy across 8 datasets spanning various domains.

Importantly, we move beyond simply evaluating the models in terms of their predictive accuracy to assess the external validity of the additional parameters. We show that they relate significantly to post-test outcomes. Validation and interpretation of statistical model parameter fits are a critical step towards successfully bridging EDM, the science of learning, and instruction.

1.1 The Additive Factors Model
AFM is a logistic regression model that extends item response theory by incorporating a growth or learning term. This statistical model (Equation 1) gives the probability p_{ij} that a student i will get a problem step j correct based on the student's baseline ability (θ_i), the baseline difficulty (β_k) of the required knowledge components or KCs on that problem step (Q_{jk}), and the improvement (γ_k) in each of the required KCs with each additional practice opportunity multiplied by the number of practice opportunities (T_{ik}) the student has had with that KC prior to the current problem step [4].

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k \in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik}) \quad (1)$$

AFM accommodates some individualization with the student ability parameter but makes the simplifying assumption that students learn at the same *rate*, since the original purpose of AFM was to refine KC models [4]. Here, we investigate whether extensions of AFM can accommodate variability in student learning rates and provide meaningful information about learning rate differences.

2. IDENTIFYING AND MODELING LEARNING RATE VARIATION
To explore adding learning rate variation to AFM, we created two new models extending AFM. The first model (AFM+StudRate) adds a per-student learning rate parameter, and the second model (AFM+GroupRate) adds a per-group learning rate parameter whereby membership among the three groups is determined using the method described in Section 2.1.

2.1 Student classification method
To classify students, we sought to identify those who improve—with each practice opportunity—more (or less) so than would be predicted by traditional AFM, which has a per-KC rate parameter that already accounts for the learning rate variability that is predicted by the KCs present at each opportunity. To do so, we examined the patterns in residual errors across opportunity counts after the data are fit with traditional AFM. A student whose learning curve is steeper than that predicted by AFM will exhibit

systematically increasing residual errors; i.e., residuals will correlate positively with opportunity count. Conversely, a student whose performance consistently increases *less* per opportunity than AFM predicts will exhibit a negative correlation between residual error and opportunity count.

To leverage this feature of residual error to classify students, we first fit the baseline AFM model to a full dataset (all students and KCs). Then, for each individual student, deviance residuals were computed, comparing the AFM model prediction against the actual data. Correlation coefficient cut-offs were set for each dataset at r > 0.1 for the "steep" learning-curve group and r < -0.1 for the "flat/declining" learning-curve group. Based on exploratory analyses, we selected the most stringent cut-off that yielded reasonable group sizes (approximately 50% students classified into either the steep or flat groups). The remaining students, whose learning curves were reasonably captured by the per-KC learning rates specified in AFM, were classified into a third "regular" group.

2.2 AFM+StudRate and AFM+GroupRate

The model that extends AFM by adding a per-student learning rate (AFM+StudRate) is given in Equation 2. It contains the parameters of traditional AFM with an additional parameter capturing the improvement (δ_i) by each student with every additional practice opportunity. Here, T_{ik} represents the practice opportunity count of a given KC required for a problem step *j*.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k\in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik} + \delta_i T_{ik}) \quad (2)$$

The model that extends AFM by adding a per-group learning rate (AFM+GroupRate) is given in Equation 3.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k\in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik} + \delta_c S_{ic} T_{ik}) \quad (3)$$

It uses the same parameters as AFM+StudRate except that each student's improvement rate with each additional practice opportunity (δ_c) is derived from a per-group rate (and thus can only take on one of three different values). Each student's group membership is specified by S_{ic}, which takes on a value of 1 when the student *i* belongs to group *c* and a value of 0 otherwise.

3. EVALUATING MODELS FOR FIT AND PREDICTIVE ACCURACY

3.1 Datasets

To test these statistical models on real educational data and to compare their predictive accuracies, we applied them across 8 datasets from DataShop [8]: Geometry Area 96-97, Cog Model Discovery Experiment Spring 2010, Cog Model Discovery Experiment Spring 2011, Cog Model Discovery Experiment Fall 2011, Assistments Math 2008-2009 Symb-DFA, Self Explanation sch_a3329ee9 Winter 2008 CL, IWT Self-Explanation Study 1 Spring 2009, and Statistical Reasoning and Practice - Fall 2009. These span a variety of content domains: Geometry, Equation solving, Story problems, English grammar, and Statistics. All of these datasets are publicly available at http://pslcdatashop.org.

We selected datasets that had already undergone significant KC model refinement via both manual and automated methods [9].

3.2 Methods

Each dataset was pre-processed based on the single-skilled KC model that achieved the best item-stratified CV performance according to values reported on DataShop. Table 1 lists the names of the KC models used and the number of KCs in each model. The three AFM models were implemented in R with student ability (θ_i), KC difficulty (β_k), and all learning rate parameters modeled as random effects, since many datasets used here were characterized by non-uniform sparsity in student-KC pairings, due to the mastery-based adaptive nature of the tutors from which the data originate. Modeling the parameters as random effects also reduces the likelihood of over-fitting the data by keeping their estimates close to zero.

The sparsity found in mastery-based datasets is particularly extreme at high opportunity counts, and this introduces noise to our classification method, which is dependent on good resolution across opportunity counts. Thus, we employed a conservative and systematic opportunity count cut-off method prior to analyses. The number of observations at each opportunity count was totaled for each student. Counts at which the average observations per student was less than 1 *and* the number of observations for any single student was 1 or fewer were excluded. In other words, at the excluded opportunity counts, no student had more than 1 *total* observation, and the majority of students did not have any. This excluded a very small percentage of total observations; the percent of observations retained are reported in the "Opp Cut-off" column of Table 1. In addition, our grouping technique required at least 5 observations in order to run the residual-by-opportunity correlations, so students who performed fewer than 5 total problem steps were excluded from the analyses. The left-most column of Table 1 reports the number of students included (with the original N in parentheses).

Models were evaluated for each dataset using Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and cross-validation measures. Two types of cross-validation (CV) were assessed: item-stratified CV, in which different random folds contain different problem steps, and student-stratified CV, in which different random folds contain different students (i.e., the model is tested on "unseen" students). Due to the random nature of the folding process, we repeated ten runs of each type of 10-fold CV, and the mean RMSEs across each run were used to compute the overall means and standard errors (in parentheses) reported in Table 2. Any CV results in which AFM+StudRate or AFM+GroupRate significantly outperforms regular AFM (as assessed by $p<0.05$ in a paired t-test between mean RMSEs across the 10 runs) are denoted with stars.

3.3 Results

The results of fitting the three statistical models to all 8 datasets are summarized in the right-most columns of Table 1.

AFM with a per-student learning rate fails to perform consistently better than regular AFM either across metrics within any dataset or across datasets. With an extra parameter per student, AFM+StudRate naturally fits training data better, but the evaluation metrics indicate over-fitting that is likely idiosyncratic (i.e., resulting in parameter estimates that will not generalize well to "unseen" items or students). Even for the AIC metric, which incorporates a smaller penalty for extra parameters than BIC, AFM+StudRate is better than regular AFM for only half of the datasets and only slightly so. By BIC, it is better than regular AFM in only one dataset. Cross-validation reveals that AFM+StudRate fails to achieve significantly lower RMSEs than regular AFM in 14 of 16 cases.

In contrast, AFM+GroupRate performs best on *all* 8 datasets by AIC, BIC, and item-stratified CV measures. It also performs the best on the majority of datasets (6 out of 8) by student-stratified CV. The superior performance according to student-stratified CV is particularly notable, because the predictions are made on data from "unseen" students. That is, no student information (not even group membership) is available for the data in the test set. The

fact that AFM+GroupRate performs better than regular AFM implies that this model is successfully capturing some student-level variability that produces better, cleaner KC parameters. This is not true for AFM+StudRate, which did not achieve significantly better student-stratified CV for any dataset.

4. RELATIONSHIP TO PRE-POST GAINS

Predictive accuracy is often used as a proxy for quality in EDM models. Assessing the validity of these student groups beyond relevance to model-fitting is equally, if not more, important. To do so, we investigated the relationship between group membership and post-test outcomes. Four of the datasets tested in Section 3 contained pre/post-test data that were accessible via DataShop: the three geometry Cog Discovery datasets and the IWT 1 dataset.

For each dataset we ran a simple regression with both pre-test score and per-group coefficients (from fitting AFM+GroupRate) as predictors of post-test score. Even after taking into account the variance explained by pre-test scores, learning rate group membership predicts post-test scores significantly for Cog Discovery Spring 2010 ($p<0.001$), Cog Discovery Fall 2011 ($p=0.016$), and Cog Discovery Spring 2011 ($p<0.001$), and marginally significantly for IWT 1 ($p=0.077$). These results suggest that group classification predicts unique variance in post-test outcomes and is thus a valid and interpretable construct.

5. DISCUSSION

5.1 Conclusions and implications

In the present work, we investigated two extensions of AFM that incorporated learning rate variation: adding a per-student learning rate parameter (AFM+StudRate) and adding a per-group learning rate parameter (AFM+GroupRate). AFM+StudRate overall did not significantly improve upon regular AFM, according to predictive accuracy metrics. In contrast, the residual-based student grouping method we developed seems to capture meaningful differences in learning rate variations. The groups have internal validity: adding a per-group learning rate to AFM improved predictive accuracy across all datasets based on the vast majority of fit metrics. They also have external validity: per-group rate

Table 1. *Dataset details and predictive accuracy metrics for each of the three statistical models fit to datasets. The percent of observations retained for analyses are shown in parentheses underneath opportunity cut-off values. Item- and student-stratified CV values are mean RMSEs over 10 separate runs of 10-fold cross validation, with standard errors in parentheses. Stars denote models with significantly better cross-validation performance (at $p<0.05$ in paired t-tests of RMSE values across CV runs) than regular AFM. The best-performing models by each metric are bolded.*

Dataset [Domain] # Students	KC Model (# KCs)	Opp Cut-off	Statistical Model	AIC	BIC	Item-Strat CV RMSE	Student-Strat CV RMSE
Geometry 1996-97 [Geometry] N = 56 (of 59)	LFASearchAIC WholeModel3 (18)	27 (99.22%)	AFM	5039.7	5072.4	.3996 (.0003)	**.4063 (.001)**
			+StudRate	5043.8	5080.5	.3991 (.0004)	.4063 (.001)
			+GroupRate	**4999.2**	**5038.4**	**.3975 (.0003)***	.4068 (.001)
Cog Discovery Spring 2010 [Geometry] N = 123 (of 123)	KTskills.Mcontext.single.sep.ind.areas (42)	80 (99.72%)	AFM	29208.5	29251.7	.3238 (.00003)	.3319 (.0001)
			+StudRate	29160.8	29221.3	.3232 (.00002)*	.3318 (.0001)
			+GroupRate	**29030.1**	**29081.9**	**.3230 (.00002)***	**.3317 (.0001)***
Cog Discovery Spring 2011 [Geometry] N = 65 (of 69)	KTracedSkills.matched.Fall2011 (7)	30 (99.3%)	AFM	4099.7	4131.5	.3877 (.0002)	.4025 (.0004)
			+StudRate	4101.4	4146.0	.3879 (.0002)	.4025 (.0004)
			+GroupRate	**4077.3**	**4115.3**	**.3856 (.0002)***	**.4017 (.0004)***
Cog Discovery Fall 2011 [Geometry] N = 103 (of 103)	KTracedSkills.Concatenated (15)	26 (97.87%)	AFM	3175.9	3208.2	.3104 (.0003)	**.3194 (.0003)**
			+StudRate	3177.8	3223.0	.3108 (.0003)	.3198 (.0003)
			+GroupRate	**3155.6**	**3194.3**	**.3090 (.0002)***	.3198 (.0003)
Assistments Symb-DFA [Story Problems] N = 318 (of 318)	Main.LFASearch Model0 (4)	11 (98.81%)	AFM	6013.1	6046.0	.4265 (.0006)	.47008 (.0001)
			+StudRate	6016.9	6062.9	.4267 (.0006)	.47008 (.0001)
			+GroupRate	**5793.2**	**5832.7**	**.4166 (.0006)***	**.47005 (.0001)**
Self-Explanation Winter 2008 [Equation Solving] N = 70 (of 71)	LFASearchAIC Model.r2 (19)	49 (98.78%)	AFM	6201.8	6235.6	.3905 (.0002)	.4140 (.0005)
			+StudRate	6201.4	6248.8	.3906 (.0002)	.4141 (.0006)
			+GroupRate	**6158.9**	**6199.5**	**.3889 (.0002)***	**.4127 (.0005)***
IWT 1 Spring 2009 [English Grammar] N = 120 (of 120)	LFASearchAIC WholeModel1 (26)	11 (98.64%)	AFM	6820.8	6854.7	.4134 (.0003)	.4392 (.0002)
			+StudRate	6815.2	6862.7	.4128 (.0003)*	.4392 (.0002)
			+GroupRate	**6752.9**	**6793.6**	**.4099 (.0002)***	**.4389 (.0002)***
Statistics Fall 2009 [Statistics] N = 52 (of 52)	LFASearchAIC Model0 (16)	30 (99.81%)	AFM	2967.8	2999.4	.3090 (.0032)	.3250 (.0003)
			+StudRate	2965.5	3009.8	.3105 (.0031)	.3250 (.0004)
			+GroupRate	**2935.5**	**2973.5**	**.3085 (.0029)***	**.3248 (.0003)***

coefficients significantly predict each group's post-test outcomes, controlling for pre-test.

Despite the focus of the AFM+GroupRate model on student-level differences, adding the per-group rate parameter produces more accurate estimates of KC parameters, based on the model's superior performance in student-stratified CV for the vast majority of datasets. The only information the model gets for fitting test data in student-stratified CV ("unseen" students whom the model has no information about with respect to ability, learning rate, or group) are the KC parameters. For this reason, AFM+GroupRate may be useful for data-driven refinement of KC parameters, which in turn has implications for instruction (e.g., parameter-setting in Knowledge Tracing based cognitive tutors [14]).

Compared to other statistical models extending AFM (Performance Factors Analysis [12], Instructional Factors Analysis [5], Recent Performance Factors Analysis [7]), AFM+GroupRate adds relatively few parameters (only three) to AFM but achieves consistent and substantive improvements in prediction. These three parameters' coefficient estimates are consistently interpretable (the per-group learning rates are ordered according to intuitions about each group's learning curve steepness), and the model avoids overloading on the interpretation of parameters.

We conducted extensive post-hoc analyses to interpret what the three learning groups actually reveal about student behavior and did not find evidence that the groups detect learning speed as an inherent trait, per se. For example, high ability students did not tend to be in the "steep" group, and low ability students did not tend to be in the "flat" group. Rather, the amount of improvement per opportunity seems to differ, more generally, depending on where the learner is on his/her *true* learning curve for any given skill. That is, the improvement per opportunity may be different for the earliest opportunities on a skill than for much later opportunities on a skill. Different students' learning curves within cognitive tutor data may vary because they start using the cognitive tutor at different points of their true learning curves for any given skill, depending on their experience with that skill prior to tutor use. We found evidence supporting this notion in post-hoc analyses. Considered in conjunction with the lack of evidence for a per-student learning rate, our findings contradict the intuitive notion that some students naturally learn faster than others.

5.2 Limitations and future work

The present results somewhat conflict with a finding from [15] that adding a per-student learning rate parameter to BKT yields substantial improvements in model fit, though we note that that report did not provide an interpretation nor any external validity evidence. We did not observe a benefit when adding a per-student learning rate parameter to AFM. Further work to compare these per-student parameter estimates across AFM and BKT and to externally validate the estimates from individualized BKT will provide insight into this issue.

Based on our post-hoc analyses, classification into the "flat/declining" group seems to capture high-ability students who descend into noisy performance at late opportunity counts (indicating boredom and/or "gaming the system" [2]) and low-ability students who never seem to improve ("wheel spinners" [1]). It would be interesting to validate this by seeing whether the detectors in [1] and [2] yield the same students when tested within the present datasets.

Another avenue for future investigation is to assess the degree to which different learning rate groups would benefit optimally from different KC models, via KC model search (as in [13]).

6. ACKNOWLEDGMENTS

We thank Carnegie Learning, Inc. for providing the majority of the datasets that were analyzed, Christopher MacLellan for insightful discussion, IES for support to RL (training grant #R305B110003), and NSF for support to LearnLab (#SBE-0836012).

7. REFERENCES

[1] Beck, J.E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. AIED, 431-440.

[2] Baker, R.S.J.d., Corbett, A.T., Roll, I., & Koedinger, K.R. (2008). Developing a generalizable detector of when students game the system. User Modeling and User-Adapted Interaction, 18(3), 287-314.

[3] Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., & Graesser, A.C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. International Journal of Human-Computer Studies, 68(4), 223-241.

[4] Cen, H., Koedinger, K.R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. Intelligent Tutoring Systems, 164-175.

[5] Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., & VanLehn, K. (2011). Instructional factors analysis: A cognitive model for multiple instructional interventions. 4th International Conference on EDM.

[6] Corbett, A.T., & Anderson, J.R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.

[7] Galyardt, A., & Goldin, I. M. (accepted). Move your lamp post: Recent data reflects learner knowledge better than older data. Journal of Educational Data Mining.

[8] Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

[9] Koedinger, K.R., McLaughlin, E.A., & Stamper, J.C. (2012). Automated Student Model Improvement. 5th International Conference on EDM.

[10] Lee, J.I., & Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. 5th International Conference on EDM.

[11] Pardos, Z.A., & Heffernan, N.T. (2010). Modeling individualization in a bayesian networks implementation of knowledge tracing. User Modeling, Adaptation, and Personalization, 255-266.

[12] Pavlik, P.I., Cen, H., & Koedinger, K.R. (2009). Performance factors analysis–a new alternative to knowledge tracing. AIED, 531–538.

[13] Rafferty, A.N., & Yudelson, M. (2007). Applying learning factors analysis to build stereotypic student models. Frontiers in Artificial Intelligence and Applications, 158, 697.

[14] Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. (2007). The Cognitive Tutor: Applied research in mathematics education. Psychonomics Bulletin & Review, 14(2), 249-255.

[15] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. (2013). Individualized bayesian knowledge tracing models. AIED, 171-180.

Evaluating The Relevance of Educational Videos using BKT and Big Data

Zachary MacHardy
UC Berkeley
354 Hearst Memorial Mining Building
Berkeley, CA 94720
zmmachar@cs.berkeley.edu

Zachary A. Pardos
UC Berkeley
4641 Tolman Hall
Berkeley, CA 94720
zp@berkeley.edu

ABSTRACT
Along with the advent of MOOCs and other online learning platforms such as Khan Academy, the role of online education has continued to grow in relation to that of traditional on-campus instruction. Rather than tackle the problem of evaluating large educational units such as entire online courses, this paper approaches a smaller problem: exploring a framework for evaluating more granular educational units, in this case, short educational videos. We have chosen to leverage an adaptation of traditional Bayesian Knowledge Tracing (BKT), intended to incorporate the usage of video content in addition to assessment activity. By exploring the change in predictive error when alternately including or omitting video activity, we suggest a metric for determining the relevance of videos to associated assessments. To validate our hypothesis and demonstrate the application of our proposed methods we use data obtained from both the popular Khan Academy website and two MOOCs offered by Stanford University in the summer of 2014.

Keywords
knowledge tracing, educational videos, instructional technology, bayesian inference, online education

1. INTRODUCTION
As the relative importance of MOOCs and other online learning platforms such as Khan Academy has increased, so has the importance of verifiably sound online pedagogy increased apace. While many of the lessons learned through a long history of research on the traditional classroom are applicable to the online environment, many indicators available during traditional instruction are not present for a designer of online material. In order to address the need for scalable and reproduceable evaluation, we hypothesize that by relating the use of materials and performance on subsequent assessment items, we can construct a metric to evaluate the relevance of those videos, without needing to resort to comparative studies.

To model student interactions with educational material and improvement over time, we have chosen to use an adaptation of Bayesian Knowledge Tracing (BKT), a technique developed and used with Intelligent Tutoring Systems (ITS) but which has been applied outside of that domain as well. We seek to incorporate behavior, such as video observation, which falls beyond the purview of attempting assessment items. We contrast this extended model with a simpler one excluding resource usage in order to discover whether videos contribute to model accuracy, and if some models benefit more than others.

Our ultimate goal is not to produce high predictive accuracy for the purposes of predicting students' latent knowledge, but rather to provide a quantitative framework for evaluating video resources. We set out first to prove that there is a reduction of predictive error when incorporating video resources into BKT analysis, in order to validate the inclusion of such observations. Second, we propose a metric based on a combination of both the delta in error between models using and eschewing video data and the learn rate associated with a particular video, in order to foreground both those which appear most relevant, as well as those which may need attention.

2. RELATED WORK
2.1 Bayesian Knowledge Tracing
Bayesian Knowledge Tracing [1] is used extensively in computer-assisted instruction environments, intended to approximate mastery learning. The model in its most basic form is defined by four parameters: $P(L_0)$, the prior probability that a student has mastered a particular KC, or knowledge component; $P(S)$, the probability a student who knows a concept will get an associated question wrong, or 'slip'; $P(G)$, the probability that a student who does not know a concept will correctly 'guess' the correct answer; and $P(T)$ the probability that a student who does not know a particular KC will learn it after a given observation. Through a process of Bayesian inference, an observed correct or incorrect response to an assessment item can be used to calculate a posterior probability that a student has mastered the KC. Using this posterior and $P(T)$ as described above, a new prior is calculated, accounting for the probability that the KC was learned between observations. This process is then repeated, using the updated estimate, for each subsequent observation.

We chose to use BKT as a modeling framework as it is well-studied and possesses relatively well understood properties, with parameters which are intuitively interpretable and therefore potentially actionable. Additional work has been done to extend this basic model of BKT to incorporate individualized parameters, based on factors depending both upon individual student properties (see e.g. [7], [2]), as well as properties of particular assessment items within a knowledge component [8].

Source	Total Events	Distinct KCs
Khan	353,202	176
Economics	689,709	94
Statistics	337,428	70

Table 1: Properties of the three sources

2.2 Online Course Resources

There has been a fair amount of research devoted to studying the efficacy of videos, forums, and other study aids offered in online educational contexts. Past work has typically focused on issues such as student attrition, student interaction, and building student-facing recommender systems. For example, Yang et al. described a framework for helping students sift through the the large volume of forum discussion posts in order to find content relevant to them [10]. Similar efforts have been made to provide recommendations for more general content, using methods such as social media analysis and reinforcement learning [5] [9].

Relative to the research on student perception and experience in the MOOC context, little attention has been paid to that of the instructor. That is not to say that such work has been absent. Guo et al. [3] and Kim et. al [4] offer guidance for the construction of videos used in MOOCs. Explorations of the application of Item Response theory in a MOOC environment [6] similarly offer instructors guidance in evaluating the efficacy of their assessments using traditional methods. Yousef et al. constructs an inventory of features, pedagogical and technological, which contribute to a sense of course quality. [11]. Yet there remains a relative paucity of research on the quantitative assessment of content outside of the scope of assessment items.

3. DATA

In order to demonstrate the generalizability of our results, we leveraged three sources of event log data. Two of our datasets were taken from Stanford Online courses run using the edX platform: 'Statistics and Medicine' and 'Principles of Economics.' The third was taken from the popular Khan Academy Website. See table 1 for details.

The data we obtained from Khan Academy contains observation events collected over about two years, from June 2012 to February 2014, while both edX courses were offered from June to September of 2014. Assessment items in Khan are categorized hierarchically as part of a larger 'exercise' representing a particular skill, and further as a member of a 'problem type,' describing the template used to generate a specific problem, while exercises from edX are categorized as individual problems. For the sake of simplicity we have chosen to consider each exercise as a separate knowledge component (KC) for the purposes of training BKT models.

For both the Khan and edX data, there was not an immediately available canonical mapping between videos and associated problems. By scanning the logs of learner activity and using a metric combining chronological proximity of use as well as frequency of associated observation, we produced a mapping between videos and their related KCs. Because our goal was not to produce a generative procedure for semantically associating log events, we chose our method to be sufficiently successful without introducing unnecessary complexity. However, this does introduce possible sources of error in terms of both overlooked and spuriously constructed mappings.

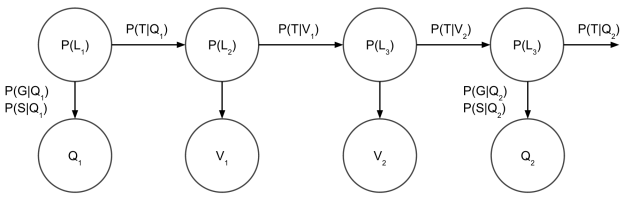


Figure 1: The Template-Videos Model

4. METHODS

Though the previous section describes the fundamentals of Bayesian Knowledge Tracing, we employ several extensions to the model. First, and for all models used in evaluation, we condition $P(G)$ and $P(S)$ for each observation on which specific problem template is observed, to model varying template difficulty. We will refer to this model as 'Standard BKT'.

Second, we similarly condition the transition probability $P(T)$ on the observed problem template, generating a second distinct but still video-free 'Template' model. We include this model for the Khan data for the sake of completeness, but note that there is only a single template for each edX problem in the data and thus the results of this extension are omitted for both the 'Statistics and Medicine' and 'Principles of Economics' cases

Third, we extend our model to incorporate video observations, conditioning $P(T)$ either on the specific template observed or the specific video, generating the 'Template Videos' model. The presence of a video observation functions similarly to that of a problem attempt, save that as there is no associated student response to be considered, a video is associated only with a unique $P(T)$. We simplify the 'Template Videos' into a fourth 'Template 1 Video' model, conditioning $P(T)$ only on the presence of either a video or a question, but not the specific identity of the resource observed.

All models were trained and evaluated using 5-fold cross validation. For each model above, one BKT model was trained for each of the knowledge components. For each model, for each fold, each of the KC models was randomly initialized and trained using Expectation Maximization (EM) algorithm to minimize the log likelihood of the observed events 25 times, with the maximally likely resulting model chosen for that model-fold-model tuple. The metric used to compare the four models is the root mean squared error (RMSE) taken across all five folds.

5. RESULTS AND DISCUSSION

Tables 2, 3, and 4 describe the results of running the data through the three analytical models. In each case, the 'Template Videos' and 'Template 1 Video' models tended to perform best, while the 'Template' model, using the Khan Academy

data, showed no significant difference from the baseline distribution. The significance test is performed across the distribution of RMSE across each of the KC models in each data-set.

Model	Mean RMSE	Significance
Pct. Correct	.4930	.0000*
Standard BKT	.3824	—-
Template	.3824	.9448
Template Videos	.3810	.0253*
Template 1 Video	.3811	.0061*

Table 2: Khan Academy

Model	Mean RMSE	Significance
Pct. Correct	.6243	.0000*
Standard BKT	.3824	—-
Template Videos	.3715	.0000*
Template 1 Video	.3716	.0000*

Table 3: Principles of Economics

Model	Mean RMSE	Significance
Pct. Correct	.5551	.0000*
Standard BKT	.3711	—-
Template Videos	.3638	.0000*
Template 1 Video	.3642	.0000*

Table 4: Statistics and Medicine

Though the tables reflect changes in RMSE aggregated over all KC models, not all models benefited evenly from the inclusion of video resources. Among the Khan data 77 of 193 KCs saw more then a trivial amount of reduction in error, while in Statistics and Medicine and Economics, the bulk of the improvement could be seen in 57 of the 94 and 44 out of 70 models, respectively. This asymmetry of improvement is an expected behavior of the system. Intuitively, in the case that a particular video resource is either not helpful or actively harmful to a student in solving a particular problem or set of problems, this would be reflected in the trained model as additional noise, leaving the overall RMSE unaffected at best.

Rather, the presence of a statistically significant, though perhaps small, decrease in predictive error in some models is indicative of the soundness of the hypothesis that considering video usage can offer useful information.

5.1 Highest and Lowest Performing Models

In order to gain an intuition for why some models were better described by the inclusion of resources, we chose to consider a selection of the best and worst performers from each data set under the 'Template-Videos' condition. By examining what properties might explain the performance of each model, we seek insight into what sort of videos appear to offer the greatest benefits to student performance.

For the highest performing models in the Khan data, the videos appeared highly relevant to their associated exercises, often demonstrating solutions in the Khan interface. For example, 'The Fundamental Theorem of Arithmetic,' explains the manipulation of a bespoke tool created for that particular exercise, showing the completion of a practice problem using that tool.

For the low performing Khan models the possible sources of error mirror the effects seen in the high performing cases. 'Scalar Matrix Multiplication' and 'Linear Inequalities', for example, present video explanation very differently than their related videos and involve customized input fields, which may have been a source of trouble.

Though the Principles of Economics and Statistics in Medicine edX courses are formatted very differently than the lessons of Khan academy, the distinctions between the best and worst models are similar. In both cases, the best videos in the data-set are, while less compellingly visually similar than the Khan examples, pointedly related to the subsequent assessments. Additionally, most of the associated assessments allowed students only one attempt, explaining the particularly strong reduction in error when including video information.

Perhaps most interesting is that one of the best predicted models is the ninth question on the final exam of the 'Statistics and Medicine' course. The content of this question is nearly identical to content of the video from a couple of weeks previous, 'Practice Interpreting Linear Regression Results.' It is therefore unsurprising to find that the video, while not explicitly grouped with the exam, is associated with a very strong learn parameter; students who sought out the video succeed significantly more often on the assessment.

Two of the videos related to the worst models in the Economics set, 'The Spending Allocation Model', and 'The Fed and the Money Supply' are both relatively long, each over fifteen minutes. Despite their length, each video dwells only briefly on the subject concerned in the assessment, spending most of their running time on other topics, with the pertinent sections easy to skip or miss. Another worst performer is one of the first videos in the course, associated with a quiz with nearly a 90% correctness rate.

Intuitively, an unhelpful video does not contribute to a predictive model, simply adding additional complexity and noise. By measuring which videos do and do not contribute constructively to predictive accuracy, it may be possible to detect which videos might be most appropriately suggested as helpful for a learner, and which need revision. In particular, such results could be useful to an instructor or course manager in navigating what to improve and what to keep when iterating on a course between offerings.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated that the inclusion of video observations in a KT model can offer information relevant to predicting student behavior, not only in one data-set, but generalizably across multiple domains. Though the effect size is small, the statistically significant decrease in error under the 'Template 1 Video' and 'Template Videos' conditions across the three data-sets considered is an encouraging sign. It is indicative that there is information to be gleaned from a learner's use of video resources. Further,

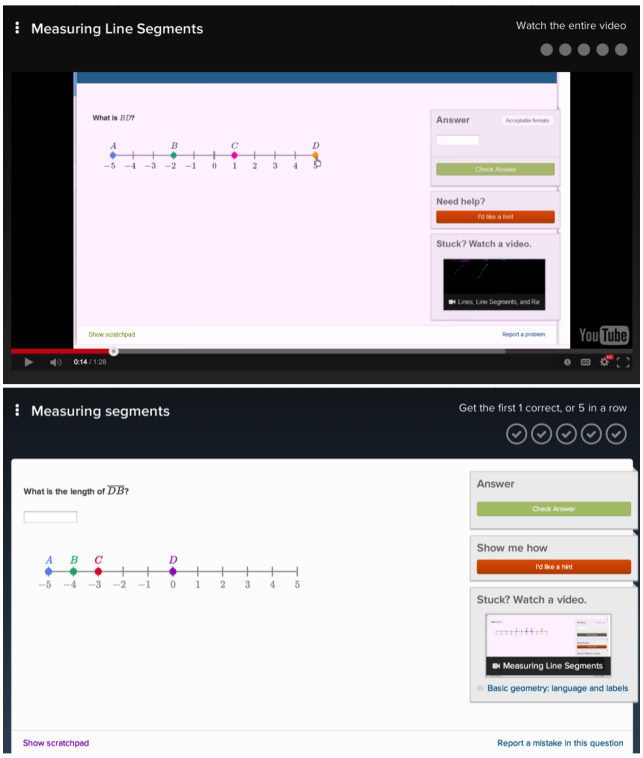


Figure 2: Videos from Khan Academy contributing maximally to model accuracy tended to closely mirror subsequent assessments

as suggested by our investigation of some of the superlative models, it is possible that the delta in error generated by a given model, coupled with the associated P(T) for a video within that model, could be a useful metric for evaluating video relevance.

One piece missing from this analysis is a canonical association of videos to exercises. Though we generated and used a set of associations, we may have lost information in the process. Another avenue worth pursuing is the possibility that some users would benefit strongly from video resources while others may not. To that end, it would be useful to examine potential reductions in error that might be made by individualizing parameters to each KC-Student pair.

An important caveat of this analysis is to note that our results do not speak to a general 'quality' of a video, and indeed that is perhaps beyond the scope of a quantitative analysis. A video rated poorly by our metrics need not necessarily be a bad video, merely unrelated or unhelpful for a subsequent assessment task. The importance of this particular property is a matter of educational policy, and thus beyond the scope of this paper. Our goal is not to supplant the role of instructor decisions in course management, only to support them.

7. REFERENCES

[1] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec. 1994.

[2] R. S. d Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[3] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 41–50, New York, NY, USA, 2014. ACM.

[4] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 31–40, New York, NY, USA, 2014. ACM.

[5] D. Kravvaris, G. Ntanis, and K. L. Kermanidis. Studying massive open online courses: recommendation in social media. In *Proceedings of the 17th Panhellenic Conference on Informatics*, pages 272–278. ACM, 2013.

[6] J. P. Meyer and S. Zhu. Fair and equitable measurement of student learning in moocs: An introduction to item response theory, scale linking, and score equating. *Research & Practice in Assessment*, 8(1):26–39, 2013.

[7] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[8] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, pages 243–254. Springer, 2011.

[9] V. Raghuveer, B. Tripathy, T. Singh, and S. Khanna. Reinforcement learning approach towards effective content recommendation in mooc environments. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on*, pages 285–289. IEEE, 2014.

[10] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.

[11] A. M. F. Yousef, M. A. Chatti, U. Schroeder, and M. Wosnitza. What drives a successful mooc? an empirical examination of criteria to assure design quality of moocs. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference On*, pages 44–48. IEEE, 2014.

Measuring Problem Solving Skills in Plants vs. Zombies 2

Valerie J. Shute
Florida State University
1114 West Call Street
Tallahassee, FL 32306
vshute@fsu.edu

Gregory R. Moore
Florida State University
1114 West Call Street
Tallahassee, FL 32306
grm13@my.fsu.edu

Lubin Wang
Florida State University
1114 West Call Street
Tallahassee, FL 32306
lw10e@fsu.edu

ABSTRACT

We are using stealth assessment, embedded in *Plants vs. Zombies 2*, to measure middle-school students' problem solving skills. This project started by developing a problem solving competency model based on a thorough review of the literature. Next, we identified relevant in-game indicators that would provide evidence about students' levels on the various problem-solving facets. Our problem solving model was implemented in the game via Bayesian networks. To validate the stealth assessment, we ran a small pilot study to collect data from students who played our game-based assessment and completed an external problem solving measure (*MicroDYN*). Preliminary results indicate that problem solving estimates derived from the game significantly correlate with the external measure, suggesting that our stealth assessment is valid. Our next steps include running a larger validation study (in progress) and developing tools to help educators interpret the results of the assessment.

Keywords

Stealth Assessment, Problem Solving, Game-Based Learning, Bayesian Networks

1. INTRODUCTION

In this paper, we describe the design, development, and preliminary validation of an assessment embedded in a video game to measure the problem solving skills of middle school students. After providing a brief background on stealth assessment and problem solving skills, we describe the game (*Plants vs. Zombies 2*) used to implement our stealth assessment, and discuss why it is good vehicle for assessing problem solving skills. Afterwards, we present the in-game indicators (i.e., gameplay evidence) of problem solving, describing how we decided on these indicators and how the indicators are used to collect data about the in-game actions of players. While discussing the indicators, we show how the evidence is used in a Bayesian network to produce an overall estimate for students' problem solving skills. We then discuss the results of a pilot validation study, which show that our stealth assessment estimate of problem solving significantly correlates with an external measure of problem solving (*MicroDYN*). We conclude with the next steps in developing the assessment and practical applications of this work.

2. BACKGROUND

2.1 Stealth Assessment

Good games are engaging, and engagement is important for learning. The challenge is validly and reliably measuring learning in games without disrupting engagement, and then leveraging that information to bolster learning. For the past 6-7 years, we have been researching various ways to embed valid assessments directly into games with a technology called *stealth assessment* (e.g., [15, 16, 20]). Stealth assessment is grounded in an assessment design framework called evidence-centered design (ECD) [10]. In general, the main purpose of any assessment is to collect information that will allow the assessor to make valid inferences about what people know, can do, and to what degree (collectively referred to as "competencies" in this paper). ECD defines a framework that consists of several conceptual and computational models that work in concert. The framework requires an assessor to: (a) define the claims to be made about learners' competencies, (b) establish what constitutes valid evidence of a claim, and (c) determine the nature and form of tasks or situations that will elicit that evidence.

Stealth assessment complements ECD by determining specific gameplay behaviors (specified in the evidence model and referred to as indicators) and linking them to the competency model [19]. As students interact with tasks/problems in a game during the solution process (see Figure 1), they are providing a continuous stream of data (captured in a log file, arrow 1) that is analyzed by the evidence model (arrow 2). The results of this analysis are data (e.g., scores) that are passed to the competency model, which statistically updates the claims about relevant competencies in the student model (arrow 3).

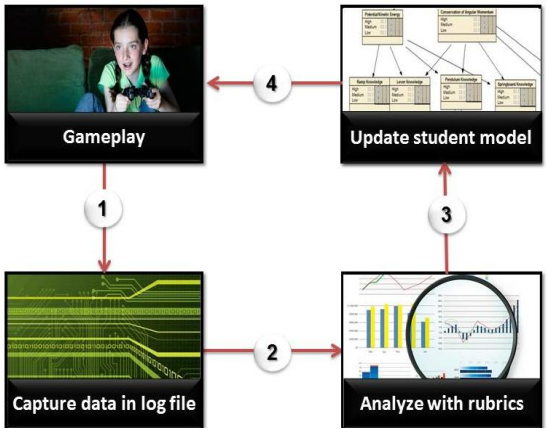


Figure 1. Stealth assessment cycle.

The ECD approach, combined with stealth assessment, provides a framework for developing assessment tasks that are explicitly

linked to claims about personal competencies via an evidentiary chain (i.e., valid arguments that connect task performance to competency estimates), and are thus valid for their intended purposes. The estimates of competency levels can also be used diagnostically and formatively to provide adaptively selected levels, feedback, and other forms of learning support to students as they continue to engage in gameplay (arrow 4). Given the dynamic nature of stealth assessment, it is not surprising that it promises advantages, such as measuring learner competencies continually, adjusting task difficulty or challenge in light of learner performance, and providing ongoing feedback.

Examples of stealth assessment prototypes, designed to measure a range of knowledge and skills—from systems thinking to creative problem solving to causal reasoning—can be found in relation to the following games: *Taiga Park* [18], *Oblivion* [20], and *World of Goo* [17], respectively. For the game *Physics Playground* (formerly Newton's Playground, see [19]), three stealth assessments were created and evaluated in relation to the validity and reliability of the assessments, student learning, and student enjoyment (see [21]). The stealth assessments correlated with associated external validated measures for construct validity and demonstrated reliabilities around .85 (i.e., using intraclass correlations among the in-game measures such as number of gold trophies received for various objects created). Furthermore, students (167 middle school students) significantly improved on an external physics test (administered before and after gameplay) despite no instruction in the game. Students also enjoyed playing the game (reporting a mean of 4 on a 5-point scale in where 1 = strongly dislike and 5 = strongly like).

Next, we briefly review our focal competency for this project—problem solving skills—and discuss the natural fit between this construct and particular video games (i.e., action, puzzle solving, simulation, and strategy games).

2.2 Problem Solving Skills

Problem solving has been studied by researchers for many decades (e.g., [3, 7, 11]). It is generally defined as any goal-directed sequence of cognitive operations [1] and is seen as one of the most important cognitive skills in any profession, as well as in everyday life [7]. Mayer and Wittrock [9] identified several characteristics of problem solving: (a) it is a cognitive process; (b) it is goal directed; and (c) the complexity (and hence difficulty) of the problem depends on one's current knowledge and skills.

In 1984, Bransford and Stein [2] integrated the collection of research at that time and came up with the IDEAL problem solving model. Each letter of IDEAL stands for an important part of the problem solving process: *Identify* problems and opportunities; *define* alternative goals; *explore* possible strategies; *anticipate* outcomes and act on the strategies; and *look* back and *learn.* Gick [4] presented a simplified model of the problem-solving process, which included constructing a representation, searching for a solution, implementing the solution, and monitoring the solution. Recent research suggests that there are two main facets of problem-solving skills: rule identification and rule application [14, 23]. "Rules" are the principles that govern the procedures, conduct, or actions in a problem-solving context. Rule identification involves acquiring knowledge of the problem-solving environment, while rule application involves controlling the environment by applying that knowledge.

Can problem solving skills be improved with practice? Polya [12] argued that people are not born with problem-solving skills. Rather, people cultivate these skills when they have opportunities to solve problems. Researchers have long argued that a central point of education should be to teach people to become better problem solvers [1, 13]. However, there is a gap between problems in formal education and those that exist in real life. Jonassen [6] noted that the problems students encounter in school are mostly well-defined, which contrasts with real-world problems that tend to be messy, with multiple possible solutions. Moreover, many problem-solving strategies that are taught in school entail a "cookbook" type of memorization and result in functional fixedness, which can obstruct students' ability to solve problems for which they have not been specifically trained. Additionally, this pedagogy can stunt students' epistemological development, preventing them from developing their own knowledge-seeking skills [8]. This is where good digital games—which have a set of goals and complicated scenarios that require the player to generate new knowledge—come in. Researchers (e.g., [22]) have argued that playing well-designed video games can promote problem-solving skills because games require constant interaction between the player and the game, usually in the context of solving many interesting and progressively more difficult problems. However, empirical research examining the effects of video games on problem-solving skills is still sparse. Our research begins to fill this gap.

3. PRESENT WORK

3.1 The Game

We are using a slightly modified version of the game *Plants vs. Zombies 2* (Popcap Games and Electronic Arts) as the vehicle for our problem solving assessment. In *Plants vs. Zombies 2* (*PvZ2*), players must plant a variety of special plants on their lawn to prevent zombies from reaching their house. Each of these plants has different attributes. For example, some plants (offensive ones) attack zombies directly, while other plants (defensive ones) slow down zombies to give the player more time to attack the zombies. A few plants generate "sun," an in-game resource needed to purchase more plants. The challenge of the game comes from determining which plants to use and where to place them in order to defeat all zombies in each level of the game.

We chose *PvZ2* as our assessment environment for two main reasons. First, we are able to alter the game because of our association with the Glasslab. Glasslab has access to the source code for *PvZ2*, so we can make direct changes to the game as needed (e.g., the particular information to be collected in the log files). This is important because it allows us to build stealth assessments directly into the game itself and to make alterations to the design of the game if needed. Second, *PvZ2* requires players to apply problem solving skills. Thus, our stealth assessment will be able to collect data relevant to problem solving and estimate learners' levels (e.g., low, medium, high) on the facets and problem solving as a whole. However, because problem solving is not easily measured, we cannot assess it directly. We instead need to define directly observable, in-game indicators of problem solving and its associated facets.

3.2 Problem Solving Model

Based on a review of the literature, we built a problem solving competency model. We divided problem solving into four facets: (a) analyzing givens and constraints, (b) planning a solution

pathway, (c) using tools and resources effectively, and (d) monitoring and evaluating progress. We then identified relevant in-game indicators of the four facets (see Section 3.3 for details). The rubrics for scoring each indicator and the statistical links between the indicators and the competency model variables comprise the evidence model. The competency and evidence models are implemented together in Bayesian networks. We created a unique Bayes net for each game level (42 total) because many indicators do not apply in every level and simple networks make computations more efficient. In the Bayes nets, the overall problem solving variable, each facet, and the associated indicators are nodes that influence each other. Each of the nodes has multiple potential states and a probability distribution that defines the likely true state of the variable. The Bayes nets accumulate data from the indicators and propagate this data throughout the network by updating the probability distributions. In this way, the indicators influence our estimates of the student's problem solving competency and its associated facets dynamically.

3.3 Indicators of Problem Solving

In line with the stealth assessment process, we defined indicators for each of the four facets of problem solving by identifying observable actions that would provide evidence per facet. This was an iterative process which began by brainstorming a large list of potential indicators. After listing all potential indicators, we evaluated each one for (a) *relevance* to their associated facets and (b) the *feasibility* of being implemented in the game. We then removed indicators that were not closely related to the facets or were too difficult or vague to implement. We repeated this process of adding, evaluating, and deleting indicators until we were satisfied with the list of indicators.

In total, there are 32 indicators for our game-based assessment: 7 for analyzing givens and constraints, 7 for planning a solution pathway, 14 for using tools and resources effectively, and 4 for monitoring and evaluating progress. Examples of indicators for each facet are shown in Table 1.

Table 1. Examples of indicators for each problem solving facet

Facet	Examples of Indicators
Analyzing Givens & Constraints	• Plants > 3 Sunflowers before the second wave of zombies arrives • Selects plants off the conveyor belt before it becomes full
Planning a Solution Pathway	• Places sun producers in the back, offensive plants in the middle, and defensive plants up front • Plants Twin Sunflowers or uses plant food on (Twin) Sunflowers in levels that require the production of X sun
Using Tools and Resources Effectively	• Uses plant food when there are > 5 zombies in the yard *or* zombies are getting close to the house (within 2 squares) • Damages > 3 zombies when firing a Coconut Cannon
Monitoring and Evaluating Progress	• Shovels Sunflowers in the back and replaces them with offensive plants when the ratio of zombies to plants exceeds 2:1

3.4 Preliminary Findings

To test the validity of the stealth assessment of problem solving skills, we recruited ten undergraduate students to play *PvZ2* for 90 minutes, as well as complete an external measure of problem solving — MicroDYN [5], a computer-based test in which participants analyzed the relationships between variables in a system and manipulated those variables to achieve a desired state. This comprised our pilot validation study. We correlated the MicroDYN scores with our stealth assessment estimates of problem solving skill to test for construct validity. The results suggest that our game-based assessment is significantly correlated with MicroDYN (r = .74, p = .03). These preliminary findings suggest that our problem solving stealth assessment is valid, but needs to be further tested with a larger sample size. We are currently running a larger validation study with 200 middle-school students and will have the results from that study in time for the EDM conference.

3.5 Limitations

There are several methodological issues with this pilot validation study. First, the sample of students was very small. Second, the participants were not from the target population of our assessment. This pilot was done with undergraduate students, but our target audience is middle school students. It is unclear if similar results will be seen with our target audience. However, middle school students do enjoy playing PvZ2 and our external measure (MicroDYN) has been successfully tested with that age group. Finally, the participants had a very limited amount of time to play the game in the small pilot study. Ninety minutes is only enough time to play about 15-20 of the game's levels. To improve the validity and reliability of the stealth assessment, players need to engage in gameplay for a longer period of time and over multiple sessions.

4. NEXT STEPS

This work is still in its early stages and we have a lot to do before it can have a meaningful impact on education. We are currently running a validation study with 200 middle school students. These students are playing *PvZ2* over three days, one hour per day. On the fourth day, the students complete MicroDYN [5] and a demographic questionnaire. For every 30 students who complete the study, we are examining the results to see if adjustments need to be made to our Bayes nets. This provides us with multiple opportunities to adjust our Bayes nets throughout the course of the validation study. Thus, this larger, ongoing study will help us to create a more valid and reliable assessment.

Our long term goal is to implement the *PvZ2* game-based assessment in middle school classrooms to help educators improve students' problem solving abilities. As part of this effort, we are teaming with Glasslab to create a dashboard that allows educators to easily interpret the results of the assessment — overall and at the individual facet level. The development of this dashboard and other tools to aid the game's implementation will occur alongside our ongoing validation study.

This focus on the validity and practicality of our game-based problem solving assessment makes it much more likely that the assessment will be both accurate and useful in classroom settings. Students can be assessed on problem solving, a key cognitive skill, in an engaging environment that presents rich problem solving situations and can parse complex patterns of students'

actions. Teachers get a valuable tool that will allow them to pinpoint students' abilities in various aspects of problem solving and, in turn, help each student improve their problem solving skills. These benefits stem from our use of evidence-centered design, which gives a framework for creating valid assessments, and stealth assessment, which gives us the ability to invisibly embed such assessments into complex learning environments such as games. By embracing evidence-centered design and stealth assessment, other researchers can also create complex and engaging assessments that meet their specific needs.

5. ACKNOWLEDGMENTS

We would like to thank our colleagues at GlassLab for supporting our work assessing problem solving in Plants vs. Zombies 2—specifically Jessica Lindl, Liz Kline, Michelle Riconscente, Ben Dapkiewicz, and Michael John. We also thank Weinan Zhao for his great programming assistance, as well as Sam Greiff and Katarina Krkovic for letting us use *MicroDYN*.

6. REFERENCES

[1] Anderson, J. R. 1980. *Cognitive psychology and its implications.* Freeman, New York, NY.

[2] Bransford, J. and Stein, B.S. 1984. *The IDEAL problem solver: A guide for improving thinking, learning, and creativity.* W. H. Freeman, New York, NY.

[3] Gagné, R. M. 1959. Problem solving and thinking. *Annual Review of Psychology*. 10, 147-172.

[4] Gick, M. L. 1986. Problem-solving strategies. *Educational Psychologist*, 21, 99-120.

[5] Greiff, S. and Funke, J. 2009. Measuring complex problem solving: The MicroDYN approach. In *The transition to computer-based assessment: New approaches to skills assessment and implications for large-scale testing*, F. Scheuermann and J. Björnsson, Eds. Office for Official Publications of the European Communities, Luxembourg, Luxembourg, 157-163.

[6] Jonassen, D. H. 2000. Toward a design theory of problem solving. *Educational Technology Research and Development*. 48, 4, 63-85.

[7] Jonassen, D. 2003. Using cognitive tools to represent problems. *Journal of Research on Technology in Education.* 35, 3, 362-381.

[8] Jonassen, D. H., Marra, R., and Palmer, B. 2004. Epistemological development: An implicit entailment of constructivist learning environments. In *Curriculum, plans, and processes of instructional design: International perspectives*, N. M. Seel and S. Dikjstra, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, 75-88.

[9] Mayer, R. E. and Wittrock, M. C. 1996. Problem-solving transfer. *Handbook of educational psychology*, D. C. Berliner and R. C. Calfee, Eds. Macmillan Library Reference, New York, NY, 47-62.

[10] Mislevy, R. J., Steinberg, L. S., and Almond, R. G. 2003. On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives.* 1, 1, 3-62.

[11] Newell, A. and Shaw, J. C. 1958. Elements of a theory of human problem solving. *Psychological Review.* 65, 3, 151-166.

[12] Polya, G. 1945. *How to solve it: A new aspect of mathematical method.* Princeton University Press, Princeton, NJ.

[13] Ruscio, A. M. and Amabile, T. M. 1999. Effects of instructional style on problem-solving creativity. *Creativity Research Journal.* 12, 251-266.

[14] Schweizer, F., Wüstenberg, S., and Greiff, S. 2013. Validity of the MicroDYN approach: Complex problem solving predicts school grades beyond working memory capacity. *Learning and Individual Differences.* 24, 42-52.

[15] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. In *Computer games and instruction*, S. Tobias and J. D. Fletcher, Eds. Information Age Publishers, Charlotte, NC, 503-524.

[16] Shute, V. J. and Ke, F. 2012. Games, learning, and assessment. In *Assessment in game-based learning: Foundations, innovations, and perspectives*, D. Ifenthaler, D. Eseryel, and X. Ge, Eds. Springer, New York, NY, 43-58.

[17] Shute, V. J. and Kim, Y. J. 2011. Does playing the World of Goo facilitate learning? In *Design research on learning and thinking in educational settings: Enhancing intellectual growth and functioning*, D. Y. Dai, Ed. Routledge Books, New York, NY, 359-387.

[18] Shute, V. J., Masduki, I., and Donmez, O. 2010. Conceptual framework for modeling, assessing, and supporting competencies within game environments. *Technology, Instruction, Cognition, and Learning.* 8, 2, 137-161.

[19] Shute, V. J. and Ventura, M. 2013. *Measuring and supporting learning in games: Stealth assessment.* The MIT Press, Cambridge, MA.

[20] Shute, V. J., Ventura, M., Bauer, M. I., and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In *Serious games: Mechanisms and effects*, U. Ritterfeld, M. Cody, and P. Vorderer, Eds. Routledge, Taylor and Francis, Mahwah, NJ, 295-321.

[21] Shute, V. J., Ventura, M., and Kim, Y. J. 2013. Assessment and learning of qualitative physics in Newton's Playground. *The Journal of Educational Research*. 106, 423-430.

[22] Van Eck, R. 2006. Building intelligent learning games. In *Games and simulations in online learning: Research & development frameworks*, D. Gibson, C. Aldrich, and M. Prensky, Eds. Idea Group, Hershey, PA.

[23] Wüstenberg, S., Greiff, S., and Funke, J. 2012. Complex problem solving — more than reasoning? *Intelligence.* 40, 1-14.

Strategic game moves mediate implicit science learning

Elizabeth Rowe
EdGE at TERC
Cambridge, MA
elizabeth_rowe@terc.edu

Ryan S. Baker
Teachers College
Columbia University
New York, NY
baker2@tc.columbia.edu

Jodi Asbell-Clarke
EdGE at TERC
Cambridge, MA
jodi_asbell-clarke@terc.edu

ABSTRACT

Educational games have the potential to be innovative forms of learning assessment, by allowing us to not just study their knowledge but the process that takes students to that knowledge. This paper examines the mediating role of players' moves in digital games on changes in their pre-post classroom measures of implicit science learning. We applied automated detectors of strategic moves, built and validated from game log data combined with coded videos of gameplay of 69 students, to a new and larger sample of gameplay data. These data were collected as part of national implementation study of the physical science game, *Impulse.* This study compared 213 students in 21 classrooms that only played the game and 180 students in 18 classrooms in where the players' teacher used game examples to bridge the implicit science learning in the game with explicit science content covered in class. We analyzed how learning outcomes between conditions were associated with six strategic moves students made during gameplay. Three of the strategic moves observed are consistent with an implicit understanding of Newton's First Law, the other three strategic moves were not. Path analyses suggest the mediating role of strategic moves on students' implicit science learning is different between the two conditions.

Keywords

Game-based science learning; Discovery with models; Automated detectors; Predictive modeling;

1. INTRODUCTION

Digital games are garnering increasing attention as potential learning environments as the volume of research increases indicating games may foster scientific inquiry, problem-solving, and public participation in breakthrough scientific discoveries [1]. Because nearly all youth and many adults participate in Internet-based games [2], educators and researchers are trying to tap this pervasive vehicle for learning and assessment environments for the 21st century [3].

Our research group studies how games can be used to improve learning of fundamental high-school science concepts (e.g. Newton's laws of motion). Our games use popular game mechanics embedded in accurate scientific simulations so that through engaging gameplay, players are interacting with digitized versions of the laws of nature and the principles of science. We hypothesize that as players dwell in scientific phenomena, repeatedly grappling with increasingly complex instantiations of the physical laws, they build and solidify their implicit knowledge over time.

It is not our intent that these games *teach* science content explicitly, but rather that they engage the learner with scientific phenomena allow them to build their implicit understandings about these phenomena through gameplay. To measure implicit learning in games, we built automated detectors of strategies we saw players using in the games [4, 5]. Thus, we address the question: *Do learners' strategic moves in the game correspond to increased implicit understanding of the science content outside the game?*

We also examine the role of the teacher in game-based learning. As Jim Gee points out, games rely on what he refers to as the Big "G" Game – the surrounding interactions that arise because of and support the game [6]. Post-game debriefing and discussions connecting gameplay with classroom learning are critical in helping students apply and transfer learning that takes place in games [7]. Our research attempts to capture the strategies players develop during gameplay that may reveal implicit knowledge, so that we can help educators seize and leverage that implicit learning to support explicit classroom learning.

Success in this approach will result in a new way to think about game-based assessments, starting not from prescribed learning outcomes, but from watching what types of strategy development actually take place. The final step of this research, reported in this paper, is to examine the extent to which strategic moves used while playing *Impulse* mediate changes in classroom measures of students' understanding of the same science content.

2. THE GAME: *IMPULSE*

The game *Impulse* is built for the web and wireless devices. Impulse challenges players use an impulse (a click or touch on the screen) to move their ball to a goal without crashing into any other (ambient) balls on the screen. All the balls have mass and obey Newton's laws of motion. As the levels of the game increase, more ambient balls are introduced, with varying mass.

Impulse is an attempt by designers to immerse a player in what is known to physicists as a n-body simulator. We hypothesize that by having to predict the motions of the particles, and their reactions to the force imparted by the impulse, the player will build implicit knowledge of forces and motions (Figure 1) that we could measure through data mining.

The first 20 levels of the game introduce players to 4 particles of different mass, providing 5 levels of experience with each of the 4

Figure 1: Impulse game

particles; across these 5 levels, the number of particles in the game space increases from 1 to eventually 10. Beginning in Level 21, players encounter particles with different masses simultaneously. As players reach higher levels with greater numbers and variety of masses of particles, they need to "study" the particles' behavior to predict the motion of particles so that they can guide their particle to the goal, not run out of energy, and avoid collision with other particles.

3. STRATEGIC MOVES

Our research attempts to capture and automatically assess the range of strategies players develop during gameplay. We identified a set of 6 strategic moves that we observe players making in the game *Impulse* (Table 1). Three of these strategic moves are theorized to constitute evidence of implicit understandings of Newton's First Law: each particle will keep moving on its path without an impulse or force from another particle. The remaining three strategic moves reflect an understanding of the game mechanic, but are not considered strong evidence of implicit understanding of Newton's First Law.

Table 1. Strategic moves and coding definitions

Strategic Move	Coding Definition
*Float	The player particle was not acted upon for more than 1 second
Toward goal	The learner intended to move the player particle toward the goal
*Stop/slow down	The learner intended to stop or slow the motion of the player particle
*Player path clear	The learner intended to move non-player particles to keep the path of the player particle clear
Goal clear	The learner intended to move non-player particles to keep the goal clear
Buffer	The learner intended to create a buffer between the player and other particles to avoid collision

*Evidence of implicit understanding of Newton's First Law

Video data was collected from 69 high school students, to develop automated detectors of these strategies. Every click in randomly selected, three-minute video segments, one per student, was coded for these strategic moves, with every player action in these video segments coded as to which strategy it represented. Two coders coded ten videos with Kappa values exceeding 0.70 for all of these strategic moves [4, 5].

We built classifiers to infer the ground truth labels created by the video coders. For each player action a set of 66 features of that action were automatically distilled, including the time since the last player action and the distance between the player particle and goal. These features were then aggregated at the click level to map to the labels provided by the video coders [6]. Classifiers were created using J48 decision trees within RapidMiner 5.3 that mapped the student behaviors in the features distilled from the clickstream data to the training labels, cross-validating at the student level. All detectors discussed here had cross-validated Kappas between 0.51 and 0.86 and A' between 0.78 and 0.97 [6].

4. IMPLEMENTATION STUDY

Having developed these detectors of student strategic moves, we then collected a much larger data set to be able to study the relationship between in-game strategic moves, pedagogical practices, and learning outcomes. To this end, we conducted an implementation study [8] to examine the conjecture that implicit learning in game play can help prepare students for classroom learning.

Forty-two teachers were assigned to one of three groups (14 per group). Teachers could include a maximum of three sections of an individual class. Of the 42 teachers who initially agreed to participate, 23 teachers completed the study (55 percent), resulting in this final sample with complete data:

Bridge: 180 students in 18 classes in which 8 teachers incorporated game examples to bridge game play and science content

Game Only: 213 students in 21 classes in which 10 teachers encouraged students to play the game, but provided no in-class interaction around the game

Control: 108 students in 11 classes in which 5 teachers taught the science content as they normally do, without games.

Students took pre-post online assessments with six items, three dealing with Newton's First Law and three dealing with Newton's Second Law. All items were written to be answerable with an intuitive understanding of the physics concepts and were piloted with think-aloud interviews. Both assessments had a maximum of 10 points possible. Assessment scores were standardized as Z-scores and all coefficients are reported in effect sizes.

Hierarchical linear modeling of data from the 23 teachers (50 classes) shows a significant positive effect of the Bridge and Game Only groups compared to the Control group on student's post-assessment scores after accounting for pre-assessment scores [8]. This group effect, however, was significantly moderated by whether or not the class was a Honors/AP class (Figure 2). There was also a significant main effect for gender, with female students receiving lower post-scores than male students.

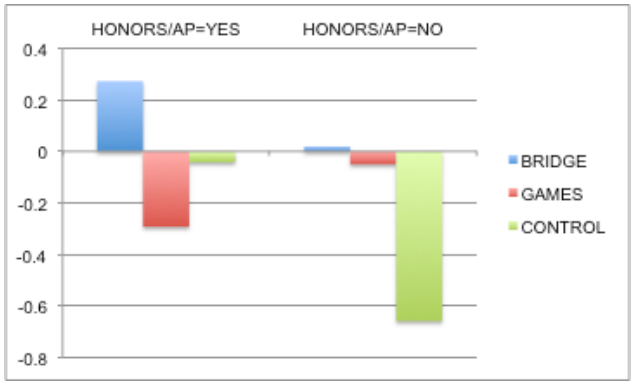


Figure 2: Predicted post-assessment scores across study conditions in Honors/AP classes versus non-Honors/AP classes (y-axis=standard deviations from the mean post-score, accounting for all components of the HLM model) [8].

The group effect was significant among students in non-Honors/AP classes. Among students in Honors/AP classes, Bridge students performed better than Game only students but not Control students. These results, while intriguing, tell us that the Bridge condition was generally best, but do not explain why Bridge was better. Did the teachers in the Bridge condition promote learning separate from the game? Or did it actually drive different behavior within *Impulse*, making the game a more valuable learning experience?

5. THE ROLE PLAYED BY IN-GAME STRATEGIC BEHAVIOR

The final step in this research, and the specific contribution novel to this paper, is to connect in-game measures of implicit science learning with external measures of those concepts. Specifically, we hypothesize that strategic moves consistent with an implicit understanding of Newton's First Law will mediate changes in these external assessments, whereas the other strategic moves will not be associated with changes in the pre-post assessments.

5.1 Apply Automated Detectors

We applied the automated detectors built with the sample of 69 students to this larger sample of gameplay data from 393 students to detect when learners used each type of strategic move. The detectors were applied to every student action during the entire duration of gameplay, 1.01 million actions in total. The same log data features were automatically distilled for this entire data set as for the initial creation of the models. Then this data was inputted into RapidMiner 5.3, along with the previously generated W-J48 decision trees model files, in order to apply the trees to the data. The result was a prediction for every click, for each of the relevant strategic moves in Table 1, of the detector's confidence that strategy was being used. Every learner action in this game was thereby annotated with an estimated probability that the learner was using each of the strategic moves.

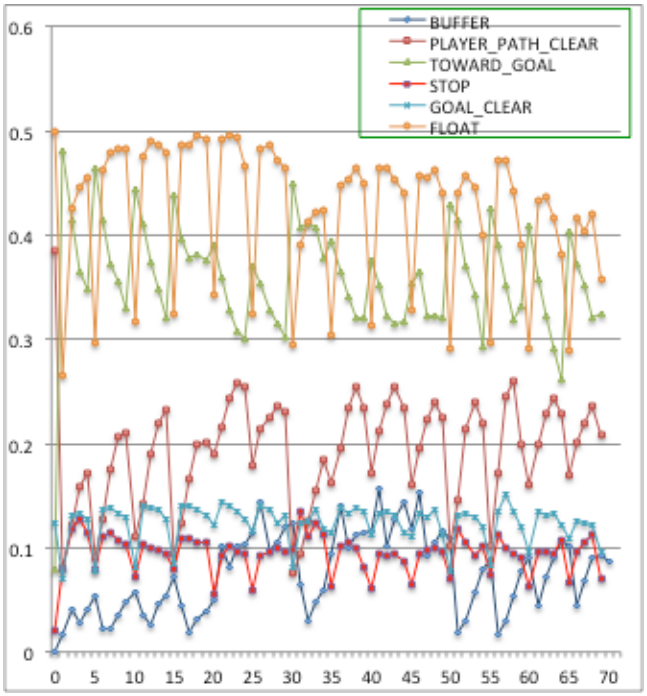


Figure 3: Average probability for each strategic move (y-axis) by game level (x-axis)

Figure 3 shows the average probability for each strategic move at each game level. The most prevalent strategic moves were Toward Goal and Float, with Float being evidence for implicit understanding of Newton's First Law. The least common strategic moves were Stop/Slow Down (evidence for implicit understanding) and Buffer. Float reflects the absence of activity (on the player particle in the time prior to the click and can co-occur with any other strategic move. Stop/Slow Down, in contrast, reflects a deliberate attempt by the player to stop or slow down the motion of the player particle. Float and Stop/Slow Down both reflect understandings of Newton's First Law e.g., a mass will keep moving until acted upon by a force, but the float strategy is a passive move and the stop strategy is an active move.

Figure 3 also shows evidence of shifts in behavior every 5 levels. The cyclical patterns in this data correspond with the planned transitions in the game. Every 5 levels, the game reduces the difficulty level of the game when a new challenge (e.g., particle with a different mass, two particles with different masses) is introduced, by decreasing the number of particles in the space (a decrease in gameplay challenge which balances for the increase in conceptual challenge). However, the reduction in the number of particles makes it more likely a player will simply push the particle toward the goal, leading to corresponding declines in all of the other strategies. Overall, as the number of particles in the game space increases, the average probability of using the simple Toward Goal strategy declines while the probabilities of using the other strategies increase.

5.2 Path Models

Path models were built to estimate the mediating role of each strategic move between prior achievement and post assessment scores using SmartPLS [9]. As pre-assessment scores and Honors/AP enrollment were significantly correlated, they were combined into a single latent variable labeled 'Prior Achievement'. Separate path models were created for the Bridge and Game Only conditions (Figures 4 and 5). The standardized coefficients appear on the paths and the adjusted R^2 values appear in the circles. T-values were calculated using a bootstrapping process with 1000 samples.

Among students in Bridge classrooms, the use of the Buffer strategy significantly mediates the impact of prior achievement and gender on post-scores (adjusted R^2 = 0.151, p=0.005). This suggests using the Buffer strategy enhanced Bridge student's understanding of the concepts, beyond what is accounted for their prior levels of achievement. In Game Only classrooms, student use of the Buffer (adjusted R2 = 0.095, p=0.018), Stop (adjusted R^2 = 0..149, p<0.001), and Float (adjusted R^2 = 0.109, p=0.031), strategic moves significantly mediate the relationship between prior achievement & gender on post-scores. In these classes with no teacher scaffolding of the gameplay, use of the Buffer and Float strategies enhanced student's understanding, but use of the Stop strategy diminished their understanding.

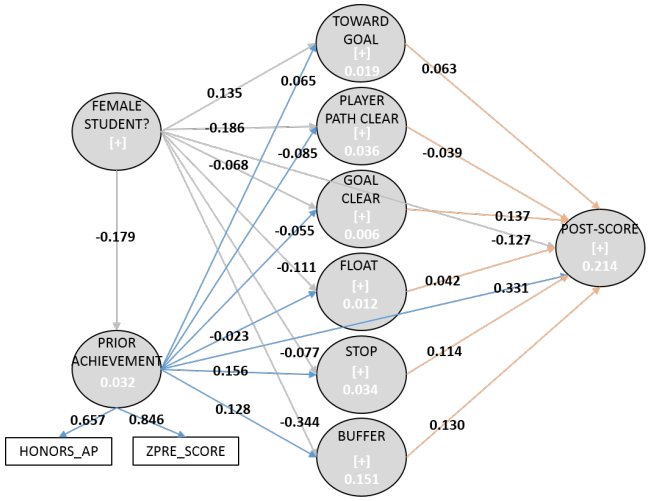


Figure 4: Full path model—Bridge Classrooms

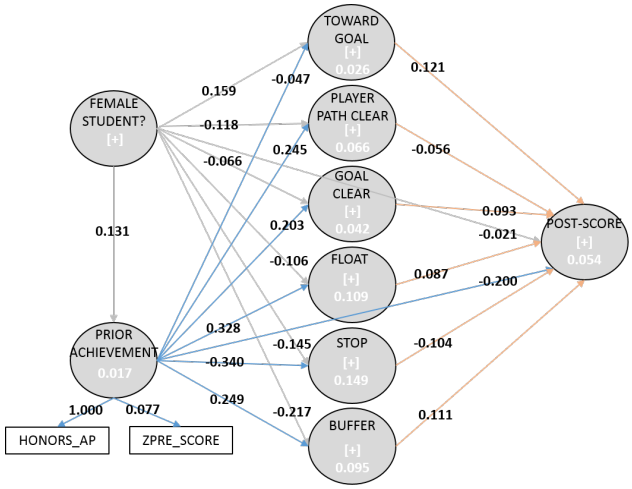


Figure 5: Full path model—Game Only Classrooms

In Bridge and Game Only classrooms, once gender differences in strategic moves are taken into account, the impact of being a Female student on post-scores is no longer significant (coeff=-0.127 in Bridge classrooms, p=0.108; coeff=-0.021 in Game Only classrooms, p=0.759). This suggests the gender main effect found in the HLM analyses may be entirely attributable to gender differences in gameplay. One potential explanation is that while females play games at equal rates as males [2], the types of games they play and the amount of time they spend doing so may vary. Success in a rapid-fire, reaction time educational game like *Impulse* may require gameplay skills more congruent with games more popular among males (e.g, first person shooters) than the social, puzzle, and role-playing games females tend to prefer [2].

6. DISCUSSION

It is noteworthy that two of the three strategies we anticipated reflecting an implicit understanding of Newton's First Law were significant mediators in Game Only classrooms. Player Path Clear, a strategic move applied to non-player particles, may not have been a significant mediator because it is likely to co-occur with Float, a strategic move applied to the player particle. By contrast, the other strategies were not significant mediators with one exception: Buffer, the simultaneous use of force on more than one particle when the particles were in close proximity to each other. Sometimes those forces were in direct opposition to the other particles (i.e., simultaneous use of the Stop strategy), while other times they were not. While Buffer was not a strategic move we a priori identified as consistent with an understanding of Newton's First Law, these results suggest it plays a mediating role similar to Stop and Float. Use of the Buffer strategy was associated with higher post-scores in Bridge and Game Only classrooms.

The negative mediating relationship of the Stop strategy in Game Only classrooms is consistent with the HLM findings shown in Figure 2, where students in Honors-AP classes did not perform on the post-assessment as well as students in non-Honors/AP classes [8]. This lack of use of the Stop strategy is consistent with the lack of understanding of Newton's Laws exhibited on the pre-post assessments. This suggests that learners who already have a basic understanding of the scientific concepts may not be aided by the game as a sole intervention. Their improvement in science understanding is enhanced when the game and the teacher bridge materials are used together. These results reinforce the importance of teachers providing bridges between gameplay and science content.

This paper also makes an important contribution to the space of problems that can be addressed by EDM. Many projects have attempted to detect strategic behavior in online learning. This project, by detecting strategic behavior explicitly connected to core concepts, and modeling how different classroom activities influence in-game behavior, shows how EDM methods can bridge understanding of the relationship between what students learn in class, and how they behave online. As such, we are able to see the concrete impact of classroom activity on gameplay behavior, and to measure its scope and manifestations.

In the long term, then, this combination of methods – automated detectors, path analysis, and classroom studies – creates the potential to make EDM useful for investigating interventions not just online, but in classroom settings as well.

7. ACKNOWLEDGMENTS

We are grateful for NSF/EHR/DRK12 grant #1119144 and our research group, EdGE at TERC, which includes Erin Bardar, Teon Edwards, Jamie Larsen, Barbara MacEachern, Katie McGrath, and Emily Kasman. Our evaluators, the New Knowledge Organization, helped establish the reliability of video coding.

8. REFERENCES

[1] Cooper, S., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature, 466*(7307), 756-760.

[2] Lenhart, A., et al. (2010) *Social Media & Mobile Internet Use Among Teens and Young Adults*. Washington, DC: Pew Internet & American Life Project.

[3] National Research Council. (2011). *Learning Science Through Computer Games and Simulations. Committee on Science Learning: Computer Games, Simulations, and Education*. M. Honey & M. Hilton (Eds.). Washington, DC: National Academies Press.

[4] Rowe, E., Baker, R., & Asbell-Clarke, J., Kasman, E., & Hawkins, W. (2014). Building automated detectors of gameplay strategies to measure implicit science learning. *Proceedings of the International Conference on Educational Data Mining*, 337-338.

[5] Rowe, E., Baker, R. & Asbell-Clarke, J. (in press). Serious games analytics to measure implicit science learning. To appear in C.S. Loh, Y. Sheng, D. Ifenhelter (Eds). *Serious Games Analytics*. New York: Springer.

[6] Gee, J. (2008). Learning and Games. In K. Salen (Ed). *The Ecology of Games: Connecting Youth, Games, and Learning*. Cambridge, MA: The MIT Press. 21–40.

[7] Ke, F. (2009). A qualitative meta-analysis of computer games as learning tools. In R. E. Furdig (Ed.), *Handbook of Research on Effective Electronic Gaming in Education* (pp. 1–32). New York: IGI Global.

[8] Rowe, E., Asbell-Clarke, J., Bardar, E., Kasman, E., & MacEachern, B. (2014, June). *Crossing the Bridge: Connecting Game-Based Implicit Science Learning to the Classroom.* Paper presented at the 10th annual meeting of Games+Learning+Society in Madison, WI.

[9] Ringle, C. M., Wende, S., & Becker, J.M. (2015). SmartPLS 3. Boenningstedt: SmartPLS GmbH, http://www.smartpls.com.

Predicting learning-related emotions from students' textual classroom feedback via Twitter

Nabeela Altrabsheh
School of Computing
Lion Terrace
University of Portsmouth
nabeela.altrabsheh@port.ac.uk

Mihaela Cocea
School of Computing
Lion Terrace
University of Portsmouth
mihaela.cocea@port.ac.uk

Sanaz Fallahkhair
School of Computing
Lion Terrace
University of Portsmouth
sanaz.fallahkhair@port.ac.uk

ABSTRACT

Teachers/lecturers typically adapt their teaching to respond to students' emotions, e.g. provide more examples when they think the students are confused. While getting a feel of the students' emotions is easier in small settings, it is much more difficult in larger groups. In these larger settings textual feedback from students could provide information about learning-related emotions that students experience. Prediction of emotions from text, however, is known to be a difficult problem due to language ambiguity. While prediction of general emotions from text has been reported in the literature, very little attention has been given to prediction of learning-related emotions. In this paper we report several experiments for predicting emotions related to learning using machine learning techniques and n-grams as features, and discuss their performance. The results indicate that some emotions can be distinguished more easily then others.

Keywords

Emotion prediction from text, Machine learning, Learning-related emotions

1. INTRODUCTION

Detecting emotions is important in the learning process [4]. Positive emotions may increase students' interest in learning, increase engagement in the classroom and motivate students [4]. Additionaly, students who are happy generally are more motivated to accomplish their learning goals.

Sentiment analysis research has grown considerably in the last decade, mainly due to the availability of rich text resources such as social networking sites, blogs and microblogs, and product reviews. Despite the name of this area, sentiment analysis is mostly focused on detection of polarity (negative or positive sentiment) rather than specific emotions. Thus, there is relatively little research on the prediction of specific emotions from text [2, 3], with even fewer reports of such research in education [9]. Moreover, from these studies (both within the educational field and outside of it), an even smaller number use machine learning to predict emotion from text, e.g. [2, 3, 9].

In this paper we focus on the prediction of emotions relevant for learning from students' textual feedback via Twitter in a classroom context using machine learning techniques. To investigate the prediction of the identified emotions from text, we experiment with several preprocessing methods, n-gram features, and machine learning techniques.

2. RELATED RESEARCH

There are four main steps to create predictive models from text with machine learning: preprocessing the data, selecting the features, applying the machine learning techniques and evaluating the results.

Preprocessing the data involves preparing the data and cleaning it from unwanted elements which may negatively affect the performance of the machine learning techniques. Some of the general preprocessing techniques used with basic text are: tokenization, convert text to lower or upper case, remove punctuation, remove numbers and, remove stop words [8].

Preprocessing Twitter data requires additional techniques due to the presence of emoticons, hashtags and chat language. Some of the Twitter-specific data preprocessing techniques from previous research [8, 11] are: removing hashtags, removing URLs, removing retweets, identifying emoticons, removing user mentions in tweets, removing Twitter special characters, and slang/chat language handling.

In relation to specific emotions detection, both general preprocessing techniques and Twitter-related preprocessing techniques have been used, e.g. removal of stop words and stemming [3], removing URLs [5], and tokenization [5].

Feature selection refers to the process of selecting relevant features for the particular prediction problem, while eliminating the features that are redundant or irrelevant. In prediction problems where the data is in the form of text, the most common features are n-grams [7]. The most commonly used n-gram for emotion detection is unigrams (one word) [7]. In contrast, there are very few studies investi-

gating the use of bigrams (two words) and trigrams (three words) in emotion prediction. However bigrams and trigrams has been used in sentiment analysis of tweets [7]. In this paper, we investigate the influence of these different n-grams and their combination on emotion detection.

Various *machine learning techniques* have been used for polarity and emotions prediction from text. In our experiments we used classifiers previously shown to work well [9]: Naive Bayes (NB), Multinomial Naive Bayes (MNB), Complement Naive Bayes (CNB), Support Vector Machines (SVM), Maximum Entropy (ME), Sequential Minimal Optimization (SMO), and Random Forest (RF).

Previous research on emotions related to learning indicates a variety of emotions experienced by learners [6]. In previous research [1], we identified from the literature a number of common emotions that are associated with learning: amused, anxiety, appreciation, awkward, bored, confusion, disappointed, embarrassed, engagement, enthusiasm, excitement, frustration, happy, motivated, proud, relief, satisfaction, shame and uninterested.

3. DATA CORPUS

The data was collected from lectures taught in English in Jordanian universities on different topics: calculus, English communication skills, database, engineering, molecular biology, chemistry, physics, science, contemporary history of the world and architecture.

Twitter was used to collect students feedback, opinions, and feelings about the lecture. For each tweet, they were asked to choose one emotion from a set of emotions provided, i.e. the 19 emotions listed in the previous section. Although tweets were used the language was formal and did not include chat language or slang, however, they did include emoticons and hashtags.

A total number of 1522 tweets were collected with their corresponding emotion label. There was one label per feedback. Some of the emotions appeared more frequently than others. The most frequent emotions that were used in our research were: Bored (336), Amused (216), Frustration (213), Excitement (178), Enthusiasm (176), Anxiety (130), Confusion (73), and Engagement (67). The least frequent ones were discarded due to insufficient data for training and testing machine learning algorithms: Happy (32), Satisfaction (31), Appreciation (26), Embarrased (18), Dissapointed (12), Uninterested (4), Proud (3), Relief (3), Shame (2), Awkward (1), and Motivated (1).

4. PREDICTION OF EMOTIONS FROM STUDENTS' FEEDBACK

Two different preprocessing levels were experimented with: (a) high preprocessing, which includes: tokenization, convert text to lower case, remove punctuation, remove numbers, remove stop words, remove hashtags, remove URLs, remove retweets, remove user mentions in tweets, and remove Twitter special characters; (b) low processing, which includes: tokenization, convert text to lower case, and remove stop words.

The high preprocessing was only used for one of the models which contained all the emotions combined, due to the low results that it led to in comparison with the low level of preprocessing for this model. Consequently, for the other models only the low preprocessing was experimented with.

The negative influence of preprocessing on the performance of the models indicates that information that is typically discarded for polarity prediction has value for the identification of specific emotions, as for example in the case of punctuation [11].

We experimented with different n-grams, i.e. unigrams, bigrams, and trigrams, and all combinations between them to find which n-gram or combination of n-grams leads to the best performance for the different models. The features that were experimented with are: Unigrams (UNI); Bigrams (BI); Trigrams (TRI); Unigrams and Bigrams combined; Unigrams and Trigrams combined; Bigrams and Trigrams combined; and Unigrams, Bigrams, and Trigrams combined.

We used the classifiers mentioned previously in section 2 due to their common use in previous research. Additionally, we used two common kernels for SVM: radial basis (RB) and linear (LIN) kernel.

We experimented with all the emotions combined and then subtracted, in turn, the emotion with the lowest number of instances. The total number of models experimented with was 16 models, which are: 7 emotions (All except engagement) + other (8 classes); 6 emotions (7 emotions except confused) + other (7 classes); 5 emotions (6 emotions except anxiety) + other (6 classes); 4 emotions (5 emotions except enthusiasm) + other (5 classes); 3 emotions (4 emotions except excitement) + other (4 classes); 2 Emotions (Amused, Bored) + other (3 classes); and each emotion + other (2 classes).

All the models were tested using 10-fold cross-validation; the accuracy and the error rate were used to assess the overall performance of the classifiers, while the precision, recall, and F-score were used to assess the ability of the classifiers to correctly identify the specific emotion(s).

The results indicate that the models with a single emotion perform better than the multi-emotion models in terms of accuracy, although one has to bare in mind that the baseline for multi-class models is lower than the baseline for 2-class models.

The results show that two classifiers performed best in terms of accuracy: the Support Vector Machine with Radial Basis kernel (RB), mainly for the 2-class models, and Sequential Minimal Optimization (SMO), mainly for the multi-class models. In term of features, unigrams and trigrams were found to lead to the best performance for the 2-class models, while unigrams combined with bigrams and trigrams led to the best performance for the multi-class models.

Despite the fact that accuracy can be useful in predicting the models performance, it does not indicate how well a classifier can predict specific emotions. As the recall indicates the percentage of correctly identified instances for a class of in-

Table 1: Highest recall for each model

Model	Tecbnique	N-gram	Accuracy	Error rate	Precision	Recall	F-score
ALL Preprocessed	ME	UNI+BI+TRI	0.32	0.68	0.34	0.33	0.33
ALL W/O Preprocessing	ME	UNI+BI	0.32	0.68	0.33	0.32	0.32
7 Emotions+ other	NB	BI+TRI	0.26	0.74	0.24	0.25	0.25
6 Emotions+ other	MNB	UNI	0.27	0.73	0.27	0.26	0.27
5 Emotions+ other	MNB	UNI+TRI	0.25	0.75	0.32	0.32	0.32
4 Emotions+ other	MNB	BI	0.26	0.74	0.29	0.38	0.33
3 Emotions + other	ME	UNI+BI+TRI	0.51	0.49	0.43	0.36	0.39
2 Emotions+ other	ME	UNI+BI+TRI	0.57	0.43	0.40	0.51	0.45
Amused	CNB	TRI	0.49	0.51	0.19	0.70	0.30
Anxiety	CNB	TRI	0.45	0.55	0.12	0.77	0.21
Bored	CNB	TRI	0.44	0.56	0.28	0.85	0.42
Confused	CNB	TRI	0.28	0.72	0.06	0.81	0.11
Engagement	CNB	TRI	0.24	0.76	0.04	0.68	0.08
Enthuisiasm	CNB	TRI	0.36	0.64	0.14	0.76	0.24
Excitement	CNB	TRI	0.37	0.63	0.15	0.86	0.26
Frustration	CNB	TRI	0.40	0.60	0.19	0.84	0.31

Table 2: Best overall models for identification of specific emotions

Model	Tecbnique	N-gram	Accuracy	Error rate	Precision	Recall	F-score
Amused	CNB	Bi+Tri	0.64	0.36	0.24	0.62	0.35
Bored	CNB	UNI+BI+TRI	0.71	0.29	0.43	0.63	0.51
Excitement	CNB	UNI+TRI	0.64	0.36	0.21	0.64	0.32

terest, it can be used to assess the ability of the classifiers to predict emotions; in addition, precision can indicate where the identification problems occur.

For most of the models with the highest accuracy, the recall is extremely low or even 0% in some cases. In addition, precision is also low for most of the models (with a few exceptions). For instance in the "engagement + other" model where the accuracy is 95% and the precision, recall, and F-score are (0-0.05)% for the emotion class. This indicates that the high accuracy is due to the correct identification of the "other" class rather than the correct identification of emotion(s).

Table 1 displays the best experimental results when focusing on the recall, i.e. the correct identification of the emotion(s). In terms of machine learning techniques, Complement Naive Bayes (CNB) performs best for half of the models, which could be explain by the ability of this technique to compensate for uneven class sizes. In terms of features, trigrams led to the best performance in the 2-class models, while unigrams combined with bigrams and trigrams led to the best performance in the multi-class models.

The fact that the models with high recall rates have low accuracy and low precision values indicates that many instances of the "other" class are wrongly classified as indicating particular emotions. In other words, although the classifiers have a higher sensitivity for the emotion classes, they are not precise in distinguishing the "other" class from the emotion class(es).

When looking at the overall picture and the balance of the evaluation metrics considered (i.e. accuracy, error rate, precision and recall), some of the models stand out – these are presented in Table 2. We found that the best classifier is Complement Naive Bayes (CNB). When looking at the features, one can notice that different combinations of n-grams led to the best performance for different classifiers. This indicates that a combination of various n-grams instead of a single n-gram is useful for the prediction of specific emotions and should be investigated further.

It is not surprising that the best performing models are for the emotions for which we had larger number of instances (see section 3), i.e. bored, amused and excitement. Interestingly, the models for excitement performed better that the ones for frustration, although there were more instances for frustration than for excitement.

From previous research studies focusing on the prediction of emotions using machine learning techniques, only one study was conducted in an educational context [9]. This research used part-of-speech (POS) tags as features, and more specifically, they experimented with the combination of the following part-of-speech tags: verb, adverb, adjective and noun. They evaluated their models using precision, recall, and F-score and found that Random Forest performed better than the other classifiers with a weighted average F-score at 0.638. Similar to our research they found that the recall score was higher than the precision. From the emotions that we identified as relevant for learning from previous literature, they only looked at anxiety, for which they obtained a precision value of 0.6 using a LogitBoot classifier. However, this re-

search was conducted on Chinese text, which has different characteristics and structures compared with English text. Moreover, the research was based on text from online chats and discussion groups. Furthermore, they used in their approach an affective words base (i.e. lexicon), where each affective word had a number associated with its degree of reflection of a particular emotion.

Outside the educational domain, there are very few studies that looked at the prediction of specific emotions from text only, which are described below.

One study, which used unigrams and a experimented with a multi-class model with 5 emotions [3], found that the Naive Bayes and Support Vector Machine classifiers performed well, leading to an accuracy of 67%. This data, however, is not representative for other types of text expressing emotions, as indicated by the low accuracy, i.e. less than 35%, of these models on test sets with other data. Similarly to the research described above, they also experimented with lexicons for specific emotions.

Another study which used unigrams as a feature and machine learning looked at predicting the presence of emotion versus the lack of emotion [2]; they obtained a maximum accuracy of 74%. However, they did not discuss the performance in terms of identifying the presence of emotion (i.e recall for the emotion). They have also used lexicons with emotion-related words.

However, very few studies investigated the use of other n-grams. Youn and Purver [10] investigated the prediction of emotions from the Chinese microblog service Sina Weibo; in their experiments they found that the models with bigrams and trigrams outperformed the models using unigrams. Similarly, our results showed that using all of the n-grams (i.e. unigrams, bigrams, and trigrams) combined led to the best identification of emotions for the multi-emotion models. Additionally, we found that trigrams led to the best identification of emotions for the 2-class models.

While it is difficult to compare the performance of our models with previous work given the variations in different experimental set-ups (e.g. data origin, language, choice of emotions, choice of features and the use of lexicons), one aspect that seems to be prevalent in previous research is the used of lexicons. Consequently, in out future work, we will investigate the use of such an affective word base for education and its effect on the prediction models.

5. CONCLUSIONS AND FUTURE WORK

In this paper we conducted several experiments with the purpose to investigate the prediction of specific emotions related to learning from students' textual classroom feedback. We focused on several learning emotions which were found to be relevant from previous literature: Amused, Anxiety, Bored, Confusion, Engagement, Enthusiasm, Excitement, and Frustration. We experimented with several preprocessing and machine learning techniques, and also with different combinations of n-gram features.

The models were evaluated using 10-fold cross-validation and using the following evaluation metrics: accuracy, error rate, precision, recall, and F-score. The best performing models were obtained for three particular emotions using 2-class models: amused, bored and excitement. The best classifier was Complement Naive Bayes (CNB). A combination in n-grams led to the best performance in most models.

In future work we will investigate the influence on prediction of a learning-related emotion lexicon; we will also investigate the relation between learning emotions and polarity.

6. REFERENCES

[1] N. Altrabsheh, M. Cocea, and S. Fallahkhair. Predicting students' emotions using machine learning techniques. In *The 17th International Conference on Artificial Intelligence in Education*, 2015. forthcoming.

[2] S. Aman and S. Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

[3] T. Danisman and A. Alpkocak. Feeler: Emotion classification of text using vector space model. In *Convention Communication, Interaction and Social Intelligence*, volume 1, pages 53–59, 2008.

[4] S. D'Mello, T. Jackson, S. Craig, et al. Autotutor detects and responds to learners affective and cognitive states. In *Workshop on Emotional and Cognitive Issues at the International Conference on Intelligent Tutoring Systems*, 2008.

[5] F. Keshtkar and D. Inkpen. A corpus-based method for extracting paraphrases of emotion terms. In *Proceedings of the NAACL HLT Workshop on Computational approaches to Analysis and Generation of emotion in Text*, pages 35–44, 2010.

[6] B. Kort, R. Reilly, and R. W. Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies*, pages 43–436. IEEE Computer Society, 2001.

[7] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, volume 10, pages 1320–1326, 2010.

[8] A. Pak and P. Paroubek. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, volume 5, pages 436–439, 2010.

[9] F. Tian, P. Gao, L. Li, W. Zhang, H. Liang, Y. Qian, and R. Zhao. Recognizing and regulating e-learners' emotions based on interactive chinese texts in e-learning systems. *Knowledge-Based Systems*, 55:148–164, 2014.

[10] Z. Yuan and M. Purver. Predicting emotion labels for chinese microblog texts. In *The 1st International Workshop on Sentiment Discovery from Affective Data*, volume 40, 2012.

[11] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan. Analyzing twitter for social TV: Sentiment extraction for sports. In *Proceedings of the 2nd International Workshop on Future of Television*, volume 2, pages 11–18, 2011.

Video-Based Affect Detection in Noninteractive Learning Environments

Yuxuan Chen
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
ychen18@nd.edu

Nigel Bosch
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
pbosch1@nd.edu

Sidney D'Mello
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
sdmello@nd.edu

ABSTRACT

The current paper explores possible solutions to the problem of detecting affective states from facial expressions during text/diagram comprehension, a context devoid of interactive events that can be used to infer affect. These data present an interesting challenge for face-based affect detection because likely locations of affective facial expressions within videos of students' faces are entirely unknown. In the current study, students engaged in a text/diagram comprehension activity after which they self-reported their levels of confusion, frustration, and engagement. Data were chosen from various locations within the videos, and texture-based facial features were extracted to build affect detectors. Varying amounts of data were used as well to determine an appropriate window of data to analyze for each affect detector. Detector performance was measured using Area Under the ROC Curve (AUC), where chance level is .5 and perfect classification is 1. Confusion (AUC = .637), engagement (AUC = .554), and frustration (AUC = .609) were detected at above-chance levels. Prospects for improving the method of finding likely positions of affective states are also discussed.

Keywords

Affect detection; facial expression recognition; reading

1. INTRODUCTION

Educational activities like playing educational games [9], interacting with a computerized tutor [4], and comprehending text [13] have been linked to affective experiences that potentially play important roles in the learning process. Thus, automatically detecting and responding to specific affective states can be a useful technique for improving educational software [5]. A wide variety of approaches have been used to detect students' emotions and tailor instruction to their affective needs (see [8] and [5] for reviews). Affect detection is a core challenge that needs to be addressed before affect-sensitive instructional strategies can be devised.

Affect detection during interactions with educational technologies are a widely studied problem. The two most common approaches involve the use of interaction data (e.g., clicks, response times) from log files (called sensor-free detection as reviewed in [1]) and the use of physiological/behavioral sensors, such as webcams, electrodermal sensors, posture sensors, and so on (called sensor-based affect detection as reviewed in [3]). As an illustrative example, Kai et al. [11] built both interaction-based and video-based affect detectors while students played an educational game called Physics Playground [14]. Their data included affect labels corresponding to specific moments in the learning session (provided by human observers in real-time). The metric of performance was A', a close approximation of Area Under the ROC Curve (AUC), where A' = .5 is chance level and 1 is perfect classification. They were able to detect affective states at levels above chance: confusion (A' = .588 for interaction-based, .622 for face-based), engaged concentration (A' = .586 interaction, .658 face), and frustration (A' = .559 interaction, .632 face).

The aforementioned study highlights two commonalities of affect detection during learning from educational software. First, the software is typically interactive in nature, thereby providing considerable opportunities for external events (e.g., a new problem, submission of a response, system feedback, a hint) to trigger affective states. Information on these events and students' responses to these events provide valuable information to guide affect detection. Second, the data (log-files, videos, etc) used to build affect detectors is accompanied by affect labels corresponding to specific moments in a learning session. This allows label-based segmentation of the data stream and affords pinpointing the sections of the data stream for affect detection (typically windows of 10-20 seconds before the labels; e.g., [9]).

Data in some educational contexts are not well suited to creating affect detectors. For example, in self-paced reading tasks there are not necessarily many key events that are likely to trigger affective responses, unlike many educational activities where there is frequent feedback and interaction. Similarly, not all educational experiences include labeled-data that can be used to pinpoint the temporal location of affective states. For example, students might self-report their affective states *after* reading an entire passage or viewing an online lecture. This raises the additional challenge of how to segment the data stream for affect detection.

The present paper involves affect detection in the context of a noninteractive, but everyday learning task, involving mechanical reasoning from illustrated texts [7]. Students were presented with a complete text passage with an associated diagram for two minutes of study. Students self-reported their affective states after each a two minute study session, rather than any specific moment in the session. This data raised many challenges. First, interaction data was non-existent as there are no page turns or other navigation features that can be used to gain information about student behaviors. Due to the lack of interaction information, we use facial features extracted from videos of students' faces to detect affective states as they processed the text/diagram. Second,

without predictable events in the task that could trigger affective states and without affect labels during the study session, the position within a video where facial expressions of affective states are likely to occur is unknown. Rather than analyzing the entire video, knowing the location of affective states is important because the duration of affective experiences can be short and the facial expressions associated with affective states can be even shorter [2,6]. To address this problem we explore affect detection using different data window sizes and window positions within face videos to determine where displays of affect tend to occur and how long they last.

We also studied the role of learning goals on affect detection performance. Specifically, students studied the illustrated texts under two different instructional conditions. The first was to simply learn about a mechanical device (general instructions). This was followed by a focused goal that either directed students to review key components of the device or to pinpoint a particular problem with the device (specific instructions). We anticipate differences in affect detection results between the two types of instructions because they are expected to engender different levels of processing. Thus, we also build separate detectors for the two types of instructional goals to determine if there was a notable difference in detection performance.

Our main approach consisted of applying machine learning techniques to build detectors of confusion, engagement, and frustration with features extracted from facial videos using CERT [12], which is a well validated computer vision tool for extracting texture-based facial features. Detection results with different window sizes and positions show both the potential and the difficulty of detecting affective states from face videos when little is known about when displays of affect might likely occur. The data in this study come from studying instructional texts with illustration, and as such is representative of potential real-world education scenarios. Thus, determining how to detect affective states in this context is important for improving computerized education systems.

2. METHOD

Data Collection. Data were collected from 88 college students from the Psychology subject pool at a large public university in the mid-South. These students from diverse backgrounds were asked to study illustrated texts about four everyday devices: an electric bell, a toaster, a car temperature gauge, and a cylinder lock. The illustrated texts were taken from Macaulay's book, *The Way Things Work* (1988), with text order counterbalanced across participants. Each of the general and specific study instructions lasted for two minutes. Videos of the students' faces were recorded with webcams mounted on the computer monitors. Upon completion of each two-minute study session, students rated their levels of engagement, confusion, and frustration on scales of 1 (very little) to 6 (very much). Students studied all four devices with device order counterbalanced across students, thereby resulting in 704 videos (88 students × 4 devices × 2 study goals per device).

Three students' videos were discarded due to recording errors, which resulted in 680 usable videos. These videos were then analyzed using CERT, which computed the likelihoods of occurrence for facial action units (AUs) in every video frame. Large outliers in AU likelihoods were found in the last two seconds of most videos, which are probably the result of students posture shifts in response to the end of the session. The last 2 seconds were removed to compensate for these anomalies, so each video was then exactly 1 minute 58 seconds long.

Feature Engineering. CERT was able to detect 20 different AUs as well as unilateral (one side of the face only) AUs, head orientation, and nose position. From the CERT data, windows of eight different sizes (2, 3, 6, 9, 12, 15, 20, and 30 seconds) were generated. For each size, windows were drawn from the beginning, middle, and end of each video. If the window came from the beginning or the end of the video, the margin from the beginning or the end was equal to the length of the window. Figure 1 illustrates examples of windows created in this manner.

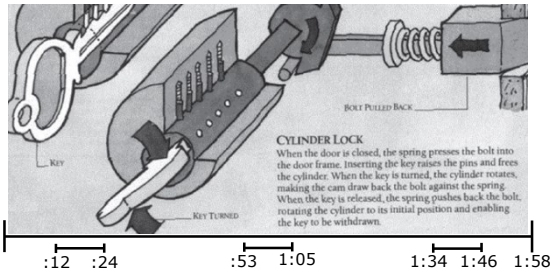


Figure 1. Positions of 12-second windows during the task.

The AU data of the windows were standardized within each student. This was followed by feature generation, in which the median, maximum, and standard deviation of the frame-level AU likelihoods were computed within each window and used as features. Some windows had less than one second of valid data, largely because the camera could not capture the student's face when they moved too much, leaned outside the camera's field of view, or when the face was occluded due to gestures. These windows were removed from the dataset, as we assumed that an affective facial expression would usually be longer than one second. Features exhibiting high multicollinearity (variance inflation factor > 5) were removed.

Supervised Classification. The features obtained above were used to construct classification models using the Waikato Environment for Knowledge Analysis (WEKA), a machine learning tool.

The classification task comprised binary high vs. low affect ratings for confusion, frustration, and boredom. The medians of the engagement, confusion, and frustration ratings on the 1-6 scale were 4, 2, and 1, respectively. We used a median split to discretize the affect ratings into "low" and "high", discarding the median instances except in the case of frustration where the median was 1. For frustration 1 was used as the "low" label.

For model validation, leave several out student-level cross-validation was applied. The training data were randomly chosen from two thirds of the students. RELIEF-F feature ranking was used to select the most diagnostic features on the training data only. The data of the remaining students were used to test the generalizability of the classifiers. Each model was trained and tested for 150 iterations with random students selected for training and testing each iteration to reduce random sampling error. Fifteen different classifiers were applied to help determine which among the eight window sizes tended to work best. Regression analysis was also explored, though the resulted models showed little promise and will not be discussed further.

3. RESULTS

The best classification models that merged videos recorded during both general and specific study instructions are listed in Table 1. The AUCs for confusion and frustration were well above chance, whereas the AUC for engagement was only slightly higher than chance level.

Table 1. Overview of results when general and specific instructional videos were combined.

Affective State	Classifier	AUC	Accuracy	No. Instances	No. Features	Window Size
Confusion	Updateable Naïve Bayes	0.637	62%	352	65	9 seconds
Engagement	AdaBoostM1	0.554	55%	403	49	20 seconds
Frustration	AdaBoostM1	0.609	64%	356	39	6 seconds

It should be noted that there were fewer than 680 instances (the total number of usable videos) for these classification models. This was largely because instances that captured less than a second of data were eliminated and the median splits that were performed to ascertain "low" and "high" values resulted in the loss of instances with affect ratings at the median.

General vs. Specific Study Instructions. The best AUCs for each video type are in Table 2. We note that for engagement, AUCs for individual general-instruction and specific-instruction models were higher than when the videos were combined. However, for confusion and frustration, it seems that the best AUCs are mostly equivalent across both individual videos and combined videos.

Table 2. Comparison of classification performance (AUC) for models using only explanation, only review, or both types of data.

Affective State	General	Specific	Both
Confusion	0.664	0.606	0.637
Engagement	0.610	0.580	0.554
Frustration	0.600	0.620	0.609

Window Position. The best AUCs (for combined models) with respect to the three window positions (i.e., beginning, middle, and end) are shown in Figure 2. Clear patterns stand out for confusion and frustration. The windows taken from the beginning of the videos seem to be more effective for confusion than those taken from the middle or the end of the videos, whereas the windows drawn from the end of the videos may best capture frustration. There is no clear pattern for engagement.

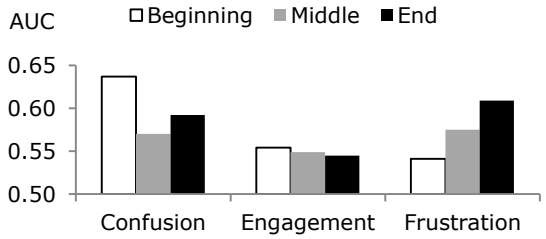


Figure 2. AUC of models using data from different positions within videos.

Window Size. Figure 3 shows the best AUCs as a function of window size for the combined models. The window position was held constant as the best window position for each affective state as noted in Figure 2. Confusion and frustration again show interesting patterns. AUC peaks at a certain window size where classification is much more successful than the surrounding window sizes. The peaks for the AUCs of confusion and frustration both occur when the window size is relatively small (9 seconds for frustration and 6 seconds for confusion). Conversely the window size seems to have no notable relationship with AUC for engagement.

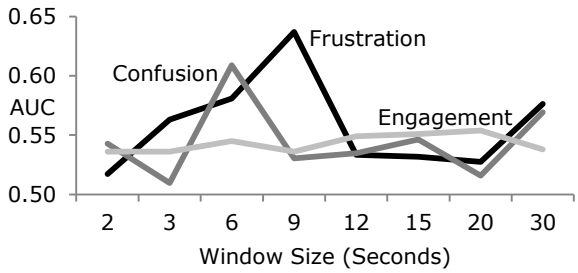


Figure 3. AUC of models as window size varies.

4. DISCUSSION

The novelty of the contributions in this paper stems from the differences between data in this study and previous affect detection work. Facial expressions of affect are often related to events in an interface (e.g., feedback, new problems), but the present study tracked affect in a noninteractive study activity – comprehension from illustrated texts. Affect labels used for detection in this study were given as retrospective judgments covering an entire 2-minute study period, so they do not provide any information about the appropriate position in the video to search for facial expressions. Thus the position of potential facial expressions in the face videos is entirely unknown. Unlike related studies with affect labels not tied to specific moments in a learning session (e.g., [10]), the current research used a subset of data from the session rather than considering all data in the session. This approach was chosen to better capture the brief nature of affective facial expressions. In the remainder of the section we discuss our main findings, and highlight limitations and avenues for future work.

Main Findings. The results above show that confusion and frustration ratings of the students can be detected with greater accuracy than the engagement rating, but that detection was successful above chance for all three affective states despite the difficulty of identifying a brief affective facial expression within the videos. However, if we split the general-instruction videos from the specific-instruction videos, the engagement rating may be better modeled, especially for the general videos. For confusion, a 9-second window at the beginning of the video worked best for classification; for frustration, a 6-second window at the end of the video was best. There were no clear patterns with respect to window position or window size for engagement.

The results suggest that when given a video with the occurrences of different affects unknown, affect ratings for confusion, frustration, and potentially engagement can still be well modeled. Smaller window sizes such as 6 or 9 seconds can be a good start to find such best models for confusion and frustration, which parallels the results in previous research [2]. Also, clips taken from the beginning of the video may yield good models for confusion, and those taken from the end of the video may work well for frustration. This seems to suggest that students' facial expressions at the beginning of the 2-minute study session can potentially indicate how confused they think they are in the end,

and that their facial features at the end of a session may provide evidence as to how frustrated they rate themselves to be. It seems that when students confront a specific task, their first impression or assessment of the difficulties and intricacies of the task can last until the end of the task. As they try to understand new concepts or to tackle problems, they experience the details of the task that they might not have known before. This may be why at the end of the task, whether they completely absorb the concepts or solve the problems, they may still feel frustrated and challenged and such emotions can be detected by analyzing facial expressions.

The reasons why engagement detection is a difficult task in this context may be due to differences in facial expressions of engagement between the general and specific study periods. It is possible that students' definitions of engagement may be linked to the particular task they are working on. General and specific study periods may be essentially different tasks, the former requiring students to intake new concepts and the latter challenging students to focus on specific aspects of concepts they have learned. Thus students may experience and display engagement differently between the two study periods, which may explain why model performance improved when each period was analyzed independently.

Limitations and Future Work. The results were promising, but there are a few limitations to this research. First, the number of videos was rather low and around 30% of the windows had to be discarded due to difficulties in registering the face (mostly due to hand-over face gestures). Also, the videos for the research were only 2 minutes long. If the window size is 30 seconds, trimming off the beginning and end 30 seconds from a video indicates that we only have one minute left for the video and the segments taken from this video can be overlapping, which is not ideal. Further research should consider a greater number of longer videos, which would allow a more thorough search of window positions and window sizes, as well as a test of the generalizability of our results to longer learning sessions.

In addition, we adopted a rather arbitrary approach of searching the start, middle, and end of each video to identify diagnostic affect expressions. In future work, we will delve more deeply into the data we already have. The feature selections of models will be examined to determine if different AUs are selected for different parts of the videos. Additionally, different methods will be applied to search for positions in the videos where affective facial expressions occur. For example, we may utilize the 9-second window size to perform a random sampling across all videos, taking segments from random positions within each video to offer more insight into how facial expressions can be leveraged for affect detection. It may also be possible to develop techniques for finding the optimal window position on a per-video basis, for example by searching for peaks or valleys in calculated features, and using windows of data specific to each video.

Concluding Remarks. In summary, this paper introduces a potential method to detect students' affective states in non-interactive instructional contexts when the locations and durations of affective facial expressions are unknown. Much work remains to be done to improve these techniques, but our results show that detecting affective states with these challenging data is certainly possible, highlighting the importance of correctly identifying the position and length of windows of data within each video.

5. ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation (NSF) (DRL 1235958) and the Bill & Melinda Gates Foundation. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF or the Bill & Melinda Gates Foundation.

6. REFERENCES

1. Baker, R. and Ocumpaugh, J. Interaction-Based Affect Detection in Educational Software. In R. Calvo, S. D'Mello, J. Gratch and A. Kappas, eds., *The Oxford Handbook of Affective Computing*. New York: Oxford University Press, 2015, 233–245.
2. Bosch, N., D'Mello, S., Baker, R., et al. Automatic detection of learning-centered affective states in the wild. *Proceedings of the 2015 International Conference on Intelligent User Interfaces (IUI 2015)*, New York, NY: ACM (2015), 379–388.
3. Calvo, R.A. and D'Mello, S. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing 1*, 1 (2010), 18–37.
4. Craig, S., Graesser, A., Sullins, J., and Gholson, B. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media 29*, 3 (2004), 241–250.
5. D'Mello, S., Blanchard, N., Baker, R., Ocumpaugh, J., and Brawner, K. I feel your pain: A selective review of affect-sensitive instructional strategies. In R. Sottilare, A. Graesser, X. Hu and B. Goldberg, eds., *Design Recommendations for Intelligent Tutoring Systems - Volume 2: Instructional Management*. 2014, 35–48.
6. D'Mello, S. and Graesser, A. The half-life of cognitive-affective states during complex learning. *Cognition & Emotion 25*, 7 (2011), 1299–1308.
7. D'Mello, S. and Graesser, A. Inducing and tracking confusion and cognitive disequilibrium with breakdown scenarios. *Acta Psychologica*, 151 (2014), 106–116.
8. D'Mello, S. and Kory, J. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. *Proceedings of the 14th ACM international conference on Multimodal interaction*, ACM (2012), 31–38.
9. Graesser, A., Chipman, P., Leeming, F., and Biedenbach, S. Deep learning and emotion in serious games. *Serious games: Mechanisms and effects*, (2009), 83–102.
10. Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N., and Lester, J.C. Automatically Recognizing Facial Expression: Predicting Engagement and Frustration. *Proceedings of the 6th International Conference on Educational Data Mining*, (2013).
11. Kai, S., Paquette, L., Baker, R., et al. Comparison of face-based and interaction-based affect detectors in physics playground. *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, International Educational Data Mining Society (in press).
12. Littlewort, G., Whitehill, J., Wu, T., et al. The computer expression recognition toolbox (CERT). *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*, (2011), 298–305.
13. Mills, C., Bosch, N., Graesser, A., and D'Mello, S. To quit or not to quit: predicting future behavioral disengagement from reading patterns. *Proceedings of the 12th International Conference on Intelligent Tutoring Systems (ITS 2014)*, Switzerland: Springer International Publishing (2014), 19–28.
14. Shute, V.J., Ventura, M., and Kim, Y.J. Assessment and learning of qualitative physics in Newton's Playground. *The Journal of Educational Research 106*, 6 (2013), 423–430.

Modeling Classroom Discourse: Do Models that Predict Dialogic Instruction Properties Generalize across Populations?

Borhan Samei[1] Andrew M. Olney[1] Sean Kelly[2] Martin Nystrand[3]
Sidney D'Mello[4] Nathan Blanchard[4] Art Graesser[1]
[1] University of Memphis [2] University of Pittsburgh [3] University of Wisconsin [4] University of Notre Dame
bsamei@memphis.edu

ABSTRACT
It has previously been shown that the effective use of dialogic instruction has a positive impact on student achievement. In this study, we investigate whether linguistic features used to classify properties of classroom discourse generalize across different subpopulations. Results showed that the machine learned models perform equally well when trained and validated on different subpopulations. Correlation-Based Feature Subset evaluation revealed an inclusion relationship between different subsets in terms of their most predictive features.

Keywords
Classroom Discourse, Machine Learning, Authenticity, Uptake

1. INTRODUCTION
Previous research on classroom instruction has shown the positive influence of dialogic instruction on student achievement [2]. Dialogic instruction is a classroom discourse strategy based on the free and open exchange of ideas between teachers and students. It is hypothesized that dialogic instruction improves achievement by increasing student engagement in classrooms [3, 5].

Previous efforts to carefully quantify teachers' use of dialogic instruction include three major studies by Nystrand and colleagues [6]. Nystrand et al.'s approach included coding discourse moves with a focus on the nature of question events, which are defined by the discourse context preceding and following a question. Question events include the question along with the response and optional evaluation/follow up. They follow a pattern that mirrors the well-known initiation response, and evaluation sequence (IRE). This coding scheme treats questions as sites of interaction and takes into account the response and evaluation. As a result, the questions alone do not uniquely determine the dialogic properties of the event; instead, they create a context through which dialogic properties may be realized.

In this research, question events were coded with five properties that were hypothesized to relate dialogic instruction to student achievement: authenticity, uptake, level of evaluation, cognitive level, and question source. However, Nystrand and Gamoran found that among these variables, authenticity and uptake were the most strongly related to student achievement [2, 8]. A question is defined as having authenticity when the asker does not have a pre-scripted answer, i.e. an open-ended question, which creates a context for students to contribute to an open ended discussion. Uptake occurs when one asks a question about something that another person has said previously. When teachers exhibit uptake, they incorporate student contributions into the discussion, potentially encouraging additional student contributions.

Question properties were live-coded by observers in Nystrand et al.'s study, a time-consuming and expensive process requiring trained classroom observers. To facilitate research into dialogic instruction, we recently developed a machine learning model to investigate the extent to which question properties can be automatically coded [9]. This previous study showed that machine learned models can predict authenticity and uptake as accurately as human experts in a setting where the questions are presented without the preceding and following context, which was the information available to the machine learned model.

Machine learned models, often referred to as *predictors or classifiers*, are sensitive to the properties of the data set on which they are trained. However, in order to perform large scale analysis, these models must be applicable to new, larger, and more diverse data. An important question in this work is whether the models systematically vary their predictions with different subpopulations in the data (e.g. different demographics). This systematic variation, essentially bias, could lead to incorrect predictions and flawed conclusions when the model is applied to a sample drawn from the same subpopulation as opposed to different subpopulations and indeed any sample where the individuals are spatially or temporally correlated may potentially have problems of generalizability.

Some recent research has focused on examining generalizability of EDM models. For example, Baker and Gowda studied the difference in student behaviors associated with disengagement in urban, suburban, and rural schools and found that urban students went off-task more often and exhibited significantly more careless behaviors than students in the rural and suburban schools [1]. Furthermore, Ocumpaugh et al, found that models trained on a population drawn primarily from one demographic grouping (rural, urban, or suburban) do not always generalize to populations drawn primarily from the other demographic groupings [7]. Generalization can sometimes occur across seemingly distinct contexts. For example, San Pedro et al. (2011) found that their models of detecting student carelessness were generalizable among different tutor interfaces (i.e. with and without an embodied conversational agent), as well as different school settings (i.e. Philippine high school and US middle school) [10].

In this paper we investigated the generalizability of two previously developed models for predicting authenticity and uptake in classroom discourse [9].

2. METHOD
We trained and tested our models using data collected from the Partnership for Literacy Study (Partnership). The data set consists of question events as recorded by the classroom observers. Partnership was a study of professional development, instruction, and literacy outcomes in middle school, in which 120 classrooms in 21 schools were observed twice in the fall and twice in the spring

over two years. The Partnership data set consists of observational data which were coded using the CLASS 4.24 computer-based data coding program [9]. Inter-rater agreement was approximately 80% on question properties with observation-level inter-rater correlations averaging approximately .95 [6].

Some of the teachers received special training in the first year and their classes were observed again in the second year. We used teacher training to split the data into Pre-training (N=7082) and Post-training (N=13655) groups. The school location was coded into categories of large and mid-size central city, urban fringe of mid-size city, small town rural outside MSA (metropolitan statistical area), and rural inside MSA. Based on the number of data points in each category, we split the data in two categories: Urban (i.e. Mid-size and Large Central City, N=13126) vs. Non-urban (the rest of categories, N=10911). Table 1 shows the distribution of authenticity and uptake across the different splits.

Table 1. Proportion of Authenticity and Uptake in different subsets and the full data set.

Category	% Authenticity	% Uptake
Non-urban : Urban	54 : 47	23 : 20
Pre-training : Post-training	39 : 52	15 : 24
Full-set	50	21

As seen in Table 1, authentic questions were more frequent than uptake in general, and the Non-urban group had higher rates of both authenticity and uptake than Urban. Overall the distribution of authenticity and uptake was similar among Non-urban, Post-training, and Full-set. Pre-training had the lowest rate of authenticity and uptake compared to others. It is also worth noting that teacher training was apparently quite effective at increasing both authenticity and uptake, as shown by the increase from Pre- to Post-training.

Based on our previous work on automating coding the questions with authenticity and uptake [9], we applied machine learning to train separate classifiers for authenticity and uptake on each of the above subsets. The models use linguistic features utilized in the classification of question types [8], including parts of speech, manually constructed bags of words (e.g., causal antecedent words), and positional information.

Most of the features are binary and indicate the presence/absence of certain keywords or part of speech tags in the question. Other features include attributes that show the position of the target keyword in the question in addition to presence/absence using four values: middle, beginning, end, and none. For example, if a question consisted of four words, e.g. "word1 word2 word3 word4" the position of "word1" is captured as beginning and "word4" as end, furthermore "word2" and "word3" are both captured as middle and if there were only two words in the question, we consider the first one as the beginning and the other as the end.

An example of a feature is causal consequent words, which include "outcomes," "results," "effects," etc. Similarly, procedural words are defined as a set of keywords including "plan," "scheme," "design," etc. Moreover, part of speech tags, such as determiner, noun, pronoun, adjective, adverb, and verb, and certain words such as "What," "How," and "Why," were also included in the feature set. More complete descriptions and justifications of these features for question classification can be found in the mentioned references.

We first trained models on each subset and evaluated their performance using 10-fold cross validation within the subset. Next, we tested generalizability by training on one subset and testing on its dual. For example, a model trained on Urban subset was tested on the Non-urban subset and vice versa. Moreover, the models trained on the full set of data were also tested on each subset. This methodology allows for the following contrasts. First, cross validation within a subset establishes a reasonable upper bound on performance since training and testing instances, while distinct, still come from the same subset. Second, training on one subset and testing on its dual subset establishes a reasonable lower bound on performance, since accuracy would be determined by shared features between the subsets rather than by distinctive properties to each subset. Training on the full data set and testing on subsets (thus training and testing on those subsets) allows similar comparisons of bias. For example, if training on the full set and testing on set A has higher accuracy than testing on set B, we may hypothesize that the features of the full model are better aligned with the features of A, or the prevalence of category distribution in the full set better matches that of A.

3. RESULTS & DISCUSSION

We first trained separate models to predict authenticity and uptake and evaluated the models using on 10-fold cross validation for each subset. For each category (e.g. Urban, Non-urban, etc.) separate decision tree models were trained and evaluated using WEKA [4]. The models for predicting uptake were trained on a random subsample of the data to obtain an even (50-50) distribution. Table 2 shows the performance of the models along with the performance of a model trained on the full set of data.

Table 2. Performance of the decision tree models trained on different data subsets using 10-fold cross validation.

Training Data	Authenticity		Uptake	
	Accuracy	*Kappa*	*Accuracy*	*Kappa*
Non-urban	0.61	0.21	0.59	0.19
Urban	0.62	0.24	0.60	0.20
Pre-training	0.64	0.24	0.61	0.23
Post-training	0.63	0.26	0.61	0.22
Full-set [9]	0.64	0.28	0.62	0.24

As seen in Table 2, the models on different splits show comparable performances, where the maximum difference on their accuracy is 0.03 (3%). To examine performance of these models and their generalizability across different subsets, we trained models on one subset and tested on its dual subset, e.g. Urban – Non-Urban. In Table 3, the performance of each model is tested on its dual. Additionally, the models trained on full set of data are tested on different subsets.

Table 3. Generalizability of models on different splits of data (trained on one tested on other).

Train	Test	Authenticity	Uptake
		Accuracy	*Accuracy*
Non-urban	Urban	0.60	0.63
Urban	Non-urban	0.62	0.62

Full-set	Non-urban	0.70	0.68
Full-set	Urban	0.68	0.68
Pre-training	Post-training	0.59	0.62
Post-training	Pre-training	0.60	0.64
Full-set	Pre-training	0.70	0.68
Full-set	Post-training	0.72	0.67

In Table 3, training on one subset and testing on its dual is never more than 2 percentage points away from the reverse. Thus the results are fairly stable. However there are several patterns of differences of interest. First, accuracy for the authenticity models when trained on Urban and tested on Non-urban is slightly higher than when trained on Non-urban and tested on Urban, however the uptake model performs slightly better when trained on Non-urban and tested on Urban than the reverse. Moreover, uptake and authenticity accuracy were higher for models trained on Post-training and tested on Pre-training compared to the reverse.

These results show that the model's performance when trained on one subset and tested on its dual is comparable to the results presented in Table 2. These results suggest that Pre-training and Non-urban are more likely to be proper subsets of Post-training and Urban respectively than the reverse. In other words, Post-training and Urban models may (by virtue of having better training data for their duals) include features that are effective on Pre-training and Urban, however this could also be due to the base rate or prevalence of authenticity and uptake in these subsets which needs further investigation.

In order to further examine the models, we compared the confusion matrices to illustrate the bias/prevalence of the models. Using the confusion matrices of models presented in Table 2 (i.e., 10-fold cross validated), we subtracted the confusion matrix when training on the Full-set from the others (Figures 1 and 2.) The resulting matrices represent the extent to which the confusion matrix of a model is different from the baseline model (i.e. Full-set). Each of the confusion matrices were separately proportionalized (before subtraction) by size of the corresponding subset to make the values comparable. Positive values in the figures indicate that the associated category occurred more often in the subset than in the Full-set. Likewise negative values mean that the category occurred less often in the subset than the Full-set.

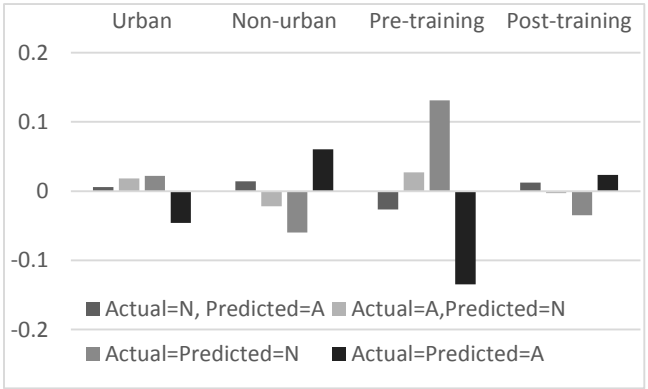


Figure 1. Normalized distance of confusion matrices of Authenticity models on subsets from full-set (A=Authenticity, N= Non-authentic).

It is seen in Figure 1 that the Urban and Post-training authenticity models are the most similar to the Full-set model because their differences with the Full-set are close to zero. This suggests that these models are not biased with respect to the Full-set. However, the Non-urban and Pre-training have larger differences with the Full-set model. Non-urban and Post-training subsets have more true-positives (Actual=Predicted=A) and less true-negatives (Actual=Predicted=N) than the Full-set while the opposite is true for Urban and Pre-training. This contrast in true-positive and true-negatives creates a trade-off in the models which previously appeared to be consistent. Specifically, Figure 1 reveals that Pre-training is more biased towards predicting N (non-authentic instances) than A (authentic instances) which may be due to the fact that there are fewer authentic instances than non-authentic in the Pre-training subset (39% vs. 50%, see Table 1). Conversely, the Non-urban model is biased towards A at the expense of N reflecting the higher distribution of A in the Non-urban subset (54% vs. 50%, see Table 1). Overall, the trade-off between true-positive and true-negative is symmetric which explains why the overall accuracy of the models is not particularly affected despite the differences in error patterns.

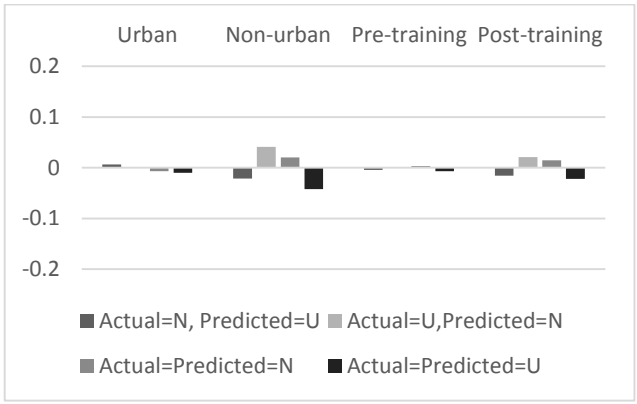


Figure 2. Normalized distance of confusion matrices of Uptake models on subsets from full-set (U=Uptake, N= Non-uptake).

Similar to Figure1, Figure 2 shows the distance between confusion matrices of uptake models. The overall distance of uptake models on subsets compared to Full-set is lower than the distance of authenticity models. Note that the uptake models were trained and 10-fold cross validated on a subsample with an even distribution (50-50) which removes the effect of prevalence on the models. Notably, the Non-urban model sacrifices more true-positives at the expense of false-negatives which explains the lower accuracy of Non-urban in predicting uptake (59% vs. 62%, see Table 2) while the rest of models are very close to the Full-set and hold a balanced tradeoff between true-positive and true-negative.

We examined the models in more detail using Correlation-Based Feature Subset evaluation (CFS). Specifically, we analyzed the frequency of each CFS feature to determine the most important CFS features for each subset. Table 4 shows the CFS results for each model. The features are presented in groups to show whether they were common between the models (shared) or exclusively included in one model only.

Table 4. CFS results, most predictive features of each model grouped based on inclusion.

Models	Authenticity	Uptake
	Urban & Non-Urban	
Shared	Wh, What	Why
Urban only	Be, Judgmental, Enablement	Neg, Pron, Causal_Antecedent
Non-urban only		Disjunction
	Pre-training & Post-training	
Shared	Judgmental, What	Neg, Metacog, Pron, Judgemental, Why
Pre-training only	Comparison	What
Post-training only	Be, Wh, Enablement	Modal, No, Causal_Antecedent

Although the models show similar performance, the most predictive features of each model is different, as seen in Table 4. However there are also marked commonalities among the groups. The features for authenticity on the Non-urban subset, for instance, are fully included in the Urban authenticity subset. Thus this analysis further supports the interpretation of inclusion suggested by the pattern of results in Table 3.

Similarly most of the features of pre-training are included in the post training features, which implies that although teachers' language changed after they received training, the result was that their linguistic behavior broadened with training such that their pre-training behavior was still evident.

4. CONCLUSION

We investigated the generalizability of previously presented models that predict authenticity and uptake in classroom discourse. Overall the results showed that the proposed models' performance is consistent among different subsets of the data set. However, we also found that some subpopulations were potentially more representative of the nature of dialogic instruction than others, making them better for classifier training.

The inclusion relationship between our subsets was investigated by comparing the confusion matrices of our models which revealed that authenticity models of supersets (i.e. Urban and Post-training) were closer to the full-set model than their duals. The consistent accuracy of the models on different subsets was attributed to the tradeoff between true-positive and true-negative predictions which was also explained by the prevalence and bias of the subsets towards one category.

We plan to apply our model to new data which is being collected currently. The proposed models will be applied with the ultimate goal of recording and coding classroom interaction in a fully automatic way and generating statistical reports to show effective instructional strategies. While the models proposed in this paper showed generalizability, another direction of future work is to improve the accuracy by adjusting current features and adding new predictive features to our models.

5. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences (R305A130030) and the National Science Foundation (IIS 1352207). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of these sponsoring agencies.

6. REFERENCES

[1] Baker, R. S., & Gowda, S. M. 2010. An Analysis of the Differences in the Frequency of Students' Disengagement in Urban, Rural, and Suburban High Schools: *Proceedings of the 3rd International Conference on Educational Data Mining*, 11-20.

[2] Gamoran, A., & Nystrand, M. 1991. Background and instructional effects on achievement in eighth-grade English and social studies: *Journal of Research on Adolescence, 1*(3), 277-300.

[3] Gamoran, A., & Nystrand, M. 1992. Taking students seriously: *Student engagement and achievement in American secondary schools*, 40-61.

[4] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The WEKA data mining software: an update: *ACM SIGKDD explorations newsletter, 11*(1), 10-18.

[5] Kelly, S. 2007. Classroom discourse and the distribution of student engagement: *Social Psychology of Education, 10*(3), 331-352.

[6] Nystrand, M., & Gamoran, A. 1997. The big picture: Language and learning in hundreds of English lessons: *Opening dialogue*, 30-74.

[7] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., & Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection: *British Journal of Educational Technology, 45*(3), 487-501.

[8] Olney, A., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. 2003. Utterance classification in AutoTutor: *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, 1-8.

[9] Samei, B., Olney, A., Kelly, S., Nystrand, M., D'Mello, S., Blanchard, N., Graesser, A. 2014. Domain Independent Assessment of Dialogic Properties of Classroom Discourse. *Proceedings of the 7th International Conference on Educational Data Mining*, 233-236.

[10] San Pedro, M. O., d Baker, R. S., & Rodrigo, M. M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics: *Artificial Intelligence in Education*, 304-311.

Breaking Off Engagement: Readers' Disengagement as a Function of Reader and Text Characteristics

Patricia J. Goedecke,[1] Daqi Dong,[1] Genghu Shi,[1] Shi Feng,[1] Evan Risko,[2]
Andrew M. Olney[1] Sidney K. D'Mello,[3] Arthur C. Graesser[1]

[1]University of Memphis
The Institute for Intelligent Systems
Memphis Tennessee, USA
1 901-678-2000
trish.goedecke@gmail.com

[2]University of Waterloo
Department of Psychology
Waterloo Ontario, Canada
1 519-888-4567
efrisko@uwaterloo.ca

[3]University of Notre Dame
Department of Psychology
Notre Dame Indiana, USA
1 574-631-322
sdmello@nd.edu

ABSTRACT
Engagement during reading can be measured by the amount of time readers invest in the reading process. It is hypothesized that disengagement is marked by a decrease in time investment as compared with the demands made on the reader by the text. In this study, self-paced reading times for screens of text were predicted by a text complexity score called formality; formality scores increase with cohesion, informational content/genre, syntactic complexity, and word abstractness as measured by the Coh-Metrix text-analysis program. Cognitive decoupling is defined as the difference between actual reading times and reading times predicted by text formality. Decoupling patterns were found to differ as a function of the serial position of the screens of text and the text genre (i.e., informational, persuasive, and narrative) but surprisingly not as a function of reader characteristics (reading speed and comprehension). This underscores the importance of mining text characteristics in addition to individual differences and task constraints in understanding engagement during reading.

Keywords

Coh-Metrix; comprehension; decoupling; engagement; formality; genre; mind wandering; reader characteristics; reading; text characteristics.

1. INTRODUCTION
Engagement during reading is essential for comprehension and learning [1]. Methods for gauging engagement include measuring time invested in the reading process and eye tracking [2-5]. We hypothesize that when mind wandering or other forms of disengagement occur, there is a marked decrease in time allocation; text characteristics then have little impact on reading times. The disjoint relationship between textual demands and time investment is termed decoupling. Cognitive decoupling is defined as the difference between actual reading times and reading times predicted by text characteristics.

This study investigates how engagement changes as a reader progresses through screens of text in moderately lengthy documents. Changes are expected to be moderated by characteristics of reader and text. Relevant reader characteristics included overall reading speed and comprehension; text characteristics included text difficulty and genre.

1.1 Text Difficulty
Text difficulty can been scaled in a variety of ways, validated by predicting grade levels of text and performance on psychometric tests of comprehension [6]. The Flesch-Kincaid Grade Level formula is a readability assessment based on word length and sentence length [7]. The Coh-Metrix tool analyzes text on multiple levels of language and discourse using computational linguistics techniques [8, 9]. Graesser et al [10] have introduced formality as a composite measure of text difficulty based on Coh-Metrix higher order principal components. Formality has a high correlation (0.72) with Flesch-Kincaid Grade Level. Discourse formality is calculated as a mean of five Coh-Metrix principal components having positive values for increasing levels of difficulty. These include: (1) referential cohesion; (2) deep (causal) cohesion; (3) informational content; (4) syntactic complexity and (5) word abstractness. Normative values (z-scores) for these 5 factors and formality are based on the TASA corpus. These norms are used to compute difficulty scores on new texts that researchers wish to analyze.

1.2 Genre and Order of Information
Genre is a discourse feature that is expected to influence engagement as well as text difficulty. Narrative texts are considered the most intrinsically engaging genre for most readers; and least difficult, compared with informational texts [6], [9], [11, 12]. Persuasive texts lie in-between narrative and informational text in expected difficulty and engagement.

The order of information presented in the text is also expected to influence engagement as well as text complexity. Readers begin engaged with a text, but may eventually lose interest and disengage as the text progresses. Research is needed to document the time allocated to texts at different points in the text. Interestingly, basic research questions have not yet been investigated at a fine grained level. Available research has only compared mind wandering as a function of texts that vary in difficulty as entire texts and these

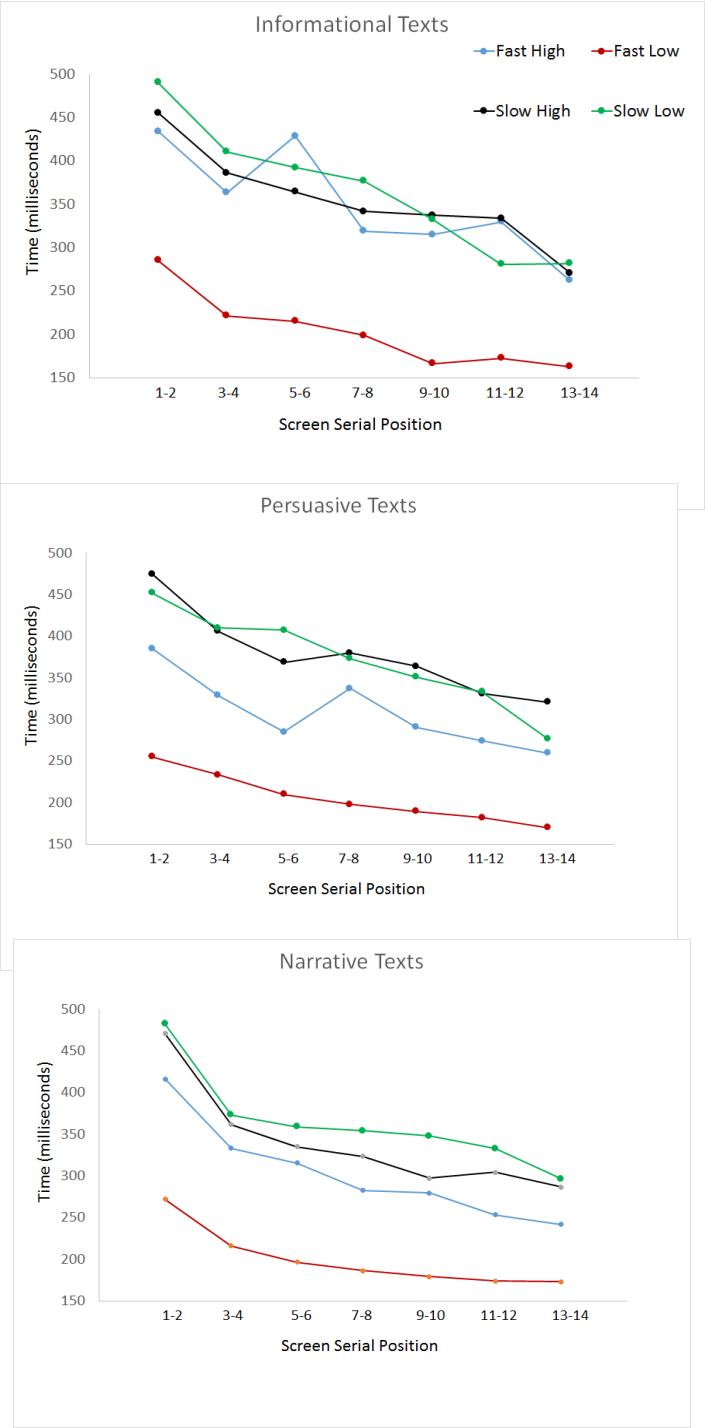


Figure 1. Reading Time per Word as a Function of Screen Serial Position, Segregated by Genre and Reader Type

studies are not consistent with respect to mind wandering increasing or decreasing with text difficulty [13].

1.3 Decoupling

Cognitive decoupling is a discrepancy between textual demands and the time a participant invests in reading a text. Decoupling increases as a function of the readers' disengagement with the text. Decoupling in this study is measured as the difference between actual reading times and times predicted by text characteristics. We interpret positive decoupling scores to indicate that a participant is investing more time in reading a text than the text characteristics demand. According to our assumptions, negative values of decoupling represent a participant investing less time than text characteristics' demands. The Coh-Metrix formality z-scores were used to measure text difficulty of a text, as normalized by the TASA corpus. Analogously, the reading time for each text segment was normalized through z-scores for individual readers on the mean reading time per word for the text segment under consideration (compared with the other text segments for that individual). Decoupling is normalized reading times for a particular person minus the normalized text difficulty based on the TASA corpus.

We predict that decoupling scores will become more negative or less positive as a reader progresses through a text, corresponding with a decrease in engagement. However, previous research [14] has not identified the shape of this decreasing function for different categories of texts and readers. These effects are predicted to be moderated by reader characteristics and genre.

2. METHODS

This study had 254 participants in two groups: 128 participated online via Mechanical Turk; 126 undergraduate Psychology students participated in a lab study.

Participants were classified according to reading time and comprehension using the Nelson Denny assessment with median split criteria. Participants read one text from each of three genres in counterbalanced order; texts assigned were randomly sampled from 24 informational, 24 persuasive, and 25 narrative texts. Following reading, participants wrote a 75-100 word summary of each text; then rated the familiarity, value, and interest for each text.

Participants used the spacebar to advance through each screen, providing reading time measurements Self-paced reading times were measured as average time per word in milliseconds for each screen of text. The number of words per screen ranged from 79 to 131, with a mean of 88.8 and a standard deviation of 11.0. The number of screens ranged from 10 to 23 per text.

3. RESULTS

3.1 Word Reading Times as Function of Text and Reader Characteristics

Mean reading times per word are presented as a function of serial position of screens of text, through position 14. Figure 1 shows times for informational (1a), persuasive (1b), and narrative texts (1c). Participants are segregated into slow versus fast readers and high versus low comprehenders.

In Figure 1, reading time functions are similar for readers with differing comprehension levels and reading speeds. We fit linear functions to each reader's times as a function of serial position, performing an ANOVA on the slopes. As expected, the slopes were negative, reflecting serial reading time decreases. A significant effect appeared in the Genre x Reading Time x Comprehension ANOVA: the slopes were lower for fast than slow readers, $F(1, 748) = 16.54$, $p < .001$. Intercepts were lower for fast readers, $F(1, 748) = 153.93$, $p < .001$. No other significant effects or interactions appeared, indicating individual differences had minimal impact on raw reading time functions. Predicted reading time per word on a page RT' follows the function: RT' (milliseconds per word) = 536 -10 * serial position (SP) of screen.

There did appear to be a dip in early serial positions and then a leveling off. Therefore we fit a quadratic equation to the reading time data. When averaging over the reader groups, the resulting

predictive equation was RT' = 409 + -23* SP + 88*SP^2. The improvement in the quadratic equation over the linear function was small when fitting curves to mean data points, R^2 = 0.97 versus 0.88, respectively. Moreover, the only coefficient that showed any differences in the Genre x Reading Time X Comprehension ANOVA was the intercept, which was lower for faster readers, F (1, 748) = 79.95, p < .001 In summary, the raw reading times showed decreases over serial position and a slight quadratic trend, but did not unveil differences in genre or individual differences.

3.2 Formality as a Function of Text Formality and Genre

It is possible that the above trends in decreasing reading times over serial position could be explained by characteristics of the text, as opposed to the readers' strategies (implicit or explicit) in allocation of reading time. We conducted an analysis of formality scores as a function of serial position, segregating the three text genres. These formality scores are plotted in Figure 2 for serial positions 1-14. The slopes for each genre were essentially flat as a function of serial position, with mean slopes of 0.00, 0.07, and 0.11 for informational, persuasive, and narrative texts, respectively. Therefore, decreasing trends in reading times cannot be attributed to systematic changes in text characteristics over serial positions.

In contrast, formality scores differed by genre, as consistent in previous studies [10]. The mean formality scores were 0.18, 0.09, and -0.26 for informational, persuasive, and narrative texts, respectively. These differences were significantly different, p < .001, showing the predicted ordering of informational > persuasion > narrative. Therefore, text characteristics varied over genre but not serial position.

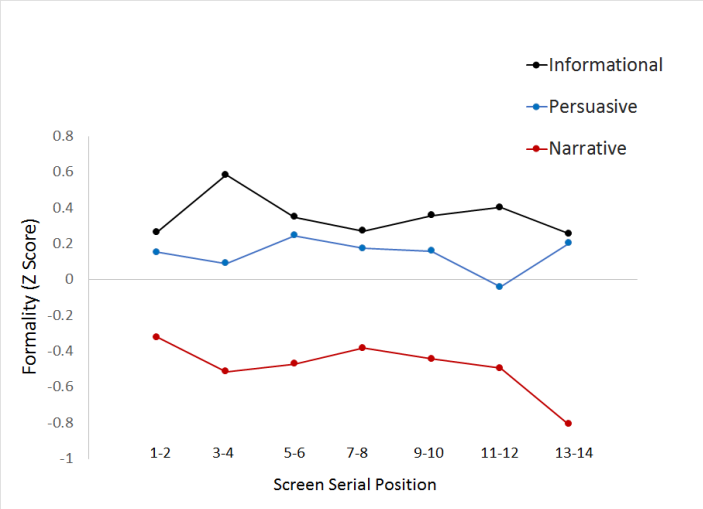


Figure 2. Formality as a Function of Screen Position, Segregated by Genre

3.3 Decoupling as a Function of Genre, Serial Position, and Reader Characteristics

It is possible that decoupling, rather than raw reading times, provides a more sensitive approach to analyzing disengagement. Figure 3 shows the decoupling scores for informational (3a), persuasive (3b), and narrative texts (3c). The participants are segregated into slow versus fast readers and high versus low comprehenders. As in the raw reading times, there did appear to be a dip in early serial positions and then a leveling off with a slow descent. The only exception was a slight upward trend for the narrative texts at the very end. When we fit a linear function to all of the participants for all of the texts, the best fit regression line yielded an R^2 =.63. A quadratic equation had a significant increase in variance explained of R^2 =.88. The best fit function was Decoupling' = 0.835 -0.204*SP + 0.010*SP^2. When we conducted a Genre x Reading Time x Comprehension ANOVA, there was only one significant effect. There was a significant effect of genre for the three coefficients in the quadratic function: F (2, 748) = 36.37; F (2, 748) = 8.46, p < .001, F (2, 748) = 11.00, all p < .001. There were no significant individual differences (reading speed or comprehension) and no interactions.

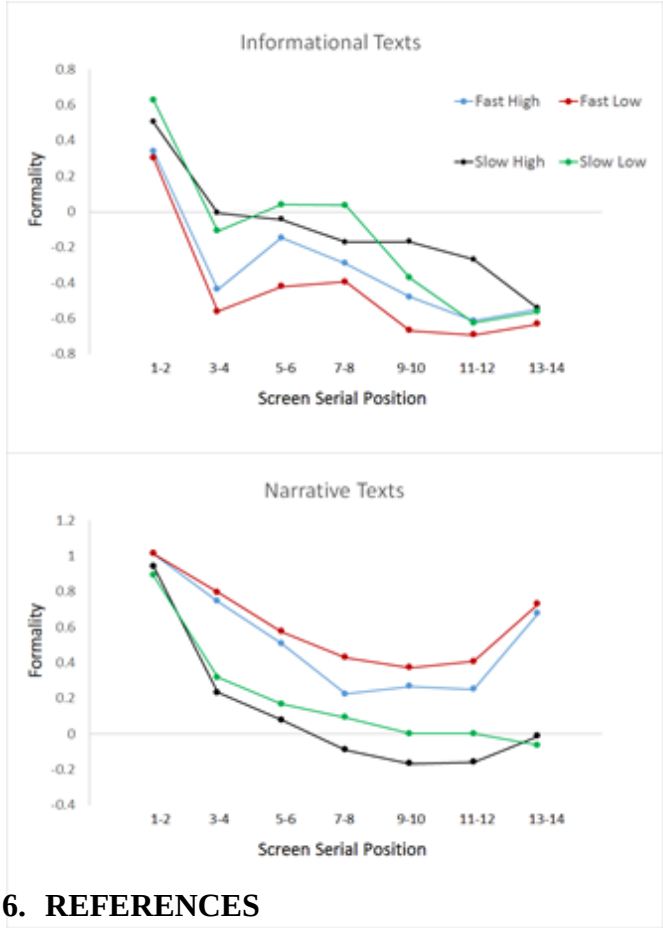
4. DISCUSSION

This study has revealed how reading times and cognitive decoupling are significantly influenced by text characteristics, namely genre and the serial position of information in the text. The pattern of results showed higher engagement (reflected in decoupling scores) in the first few screens of text and a subsequent decrease over the serial position of the screens. The deepest engagement is in the first 200-400 words, then noticeably decreases and slowly decreases thereafter (aside from an interesting upsweep for narrative texts). The quadratic function captures this trend and shows a better fit than a linear trend. It is of course strategically wise to pay attention to the early text segments because that is a critical point when the situation model is set up [11, 14], and the reader can make judgments whether the text is interesting or important to continue reading [1]. It is important to acknowledge that text difficulty is not comparatively high in early text segments, as shown in Figure 2, so increased time allocation at the beginning of a text cannot be attributed to text difficulty.

Regarding decoupling scores, text formality and difficulty show the following trend compatible with previous research using Coh-Metrix [2, 10]: informational > persuasive > narrative. However, cognitive decoupling showed the opposite ordering, such that readers tended to over allocate reading times to narrative text and under-allocate for the difficult informational text. In essence, there was a tendency to have lower engagement when the text was more difficult. The role of text difficulty has also been found to predict mind-wandering during text comprehension [13, 15] and listening to lectures [16], but the jury is still out as to (a) whether mind wandering is more prevalent in discourse that is very easy or very difficult and (b) what level of discourse analysis is most diagnostic of mind-wandering. Future research awaits an analysis of the impact on decoupling as computed via a deviation between reading time and formality and mind wandering.

5. ACKNOWLEDGMENTS

The research on was supported by the National Science Foundation (1108845) and the Institute of Education Sciences (R305C120001), Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or IES.



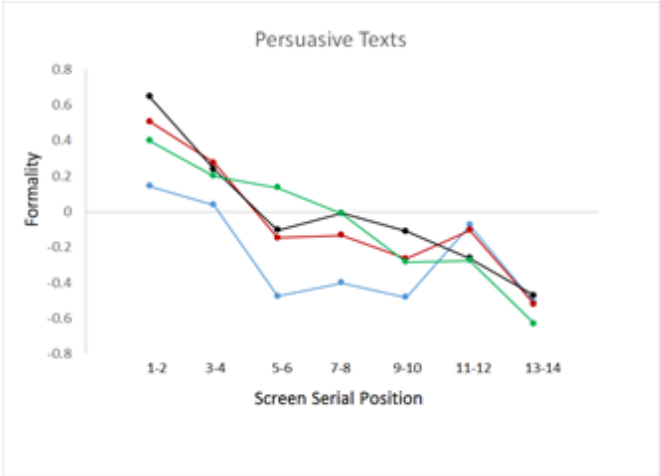


Figure 3. Decoupling by Formality as a Function of Screen Position, Segregated by Reader Type

6. REFERENCES

[1] Guthrie, J.T., S.L. Klauda, and A.N. Ho. 2013. Modeling the relationships among reading instruction, motivation, engagement, and achievement for adolescents. *Reading Research Quarterly*. 48, 1, 9-26.

[2] Franklin, M.S., J. Smallwood, and J.W. Schooler. 2011. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*. 18, 5, 992-997.

[3] Nguyen, K.-V., et al. 2014. Gotcha! Catching Kids during Mindless Reading. *Scientific Studies of Reading*. 18, 4, 274-290.

[4] Ainley, M., S. Hidi, and D. Berndorff. 2002. Interest, learning, and the psychological processes that mediate their relationship. *Journal of Educational Psychology*. 94, 3, 545.

[5] Fulmer, S.M., et al. 2015. Interest-based text preference moderates the effect of text difficulty on engagement and learning. *Contemporary Educational Psychology*. 41, 98-110.

[6] Nelson, J., et al. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. *Council of Chief State School Officers*, Washington, DC.

[7] Klare, G.R. 1974. Assessing readability. *Reading Research Quarterly*. 62-102.

[8] McNamara, D.S., et al. 2014. Automated evaluation of text and discourse with Coh-Metrix. *Cambridge University Press*.

[9] Graesser, A.C., D.S. McNamara, and J.M. Kulikowich. 2011. Coh-metrix providing multilevel analyses of text characteristics. *Educational Researcher*. 40, 5, 223-234.

[10] Graesser, A.C., et al. 2014. Coh-Metrix measures text characteristics at multiple levels of language and discourse. *The Elementary School Journal*. 115, 2, 210-229.

[11] Graesser, A.C. and D.S. McNamara. 2011. Computational analyses of multilevel discourse comprehension. *Topics in Cognitive Science*. 3, 2, 371-398.

[12] McNamara, D.S., Graesser, A. C., and Louwerse, M. M. 2013. Sources of text difficulty: Across the ages and genres. *Assessing Reading in the 21st Century: Aligning and Applying Advances in the Reading and Measurement Sciences*, J.P.S.E.A., Editor. R&L Education: Lanham, MD.

[13] Feng, S., S. D'Mello, and A.C. Graesser. 2013. Mind wandering while reading easy and difficult texts. *Psychonomic Bulletin and Review*. 20, 3, 586-592.

[14] Kintsch, W. 1998. *Comprehension: A paradigm for cognition*. Cambridge university press.

[15] Mills, C., et al. 2014. To quit or not to quit: Predicting future behavioral disengagement from reading patterns. *Intelligent Tutoring Systems*. Springer.

[16] Medimorecc, M.A., Pavlik, P., Olney, A., Graesser, A.C., and Risko, E.F. The language of instruction: Compensating for challenge in lectures. *Journal of Educational Psychology*, in press

Semantic Similarity Graphs of Mathematics Word Problems: Can Terminology Detection Help?

Rogers Jeffrey Leo John
Center for Computational Learning Systems
Columbia University
New York, NY, USA
rl2689@columbia.edu

Rebecca J. Passonneau
Center for Computational Learning Systems
Columbia University
New York, NY, USA
becky@ccls.columbia.edu

Thomas S. McTavish
Center for Digital Data, Analytics & Adaptive Learning
Pearson
Austin, TX, USA
tom.mctavish@pearson.com

ABSTRACT

Curricula often lack metadata to characterize the relatedness of concepts. To investigate automatic methods for generating relatedness metadata for a mathematics curriculum, we first address the task of identifying which terms in the vocabulary from mathematics word problems are associated with the curriculum. High chance-adjusted interannotator agreement on manual identification of math terms was achieved by considering terms in their contexts. These terms represent 13% of the vocabulary in one seventh grade mathematics text. Six classification algorithms were compared to classify math terms for this text. To avoid overfitting to this curriculum, we relied on a small number of features that exploit external knowledge sources.

1. INTRODUCTION

Curricula often lack metadata to characterize the relatedness of concepts. Our ultimate goal is to develop methods for automatic generation of knowledge graphs for mathematics from existing curricula. Towards that end, we develop a representation for math word problems that allows us to measure similarities between problems, based on the math terminology they share [14]. In this paper, we present our methods to automatically identify the math terms. While mathematics is a highly structured domain with many sources that define terms, we found no single source that captured the mathematics terms as used in the context of this curriculum. Furthermore, several terms that occur in the word problems, such as *independent*, *chances*, and *set*, are polysemous, but occur more frequently in a "mathematical" sense. We therefore annotated the full vocabulary as "math" or "non-math" based on the predominant usage in the curriculum, and found high agreement among annotators. We then tested six methods for automatic classification.

The vocabulary items to be classified were represented using a small number of features based on glossaries, web search, and corpus statistics. Only 13% of the terms in our vocabulary were labeled as "math." Such data skew is challenging for many machine learning methods. To address the class imbalance, we used ensembles of weak learners and support vector machines (SVMs), weighting errors on the "math" class more heavily. We found that SVMs were our best classifiers. The automated methods presented here can enhance existing math curricula with domain knowledge graphs of content similarity among word problems.

2. RELATED WORK

Adaptive learning environments (ALEs) have shown promising results for mathematics and other STEM subjects [18, 5, 1], even when compared with human tutors [24]. For ALE's, the domain model is typically created anew but automated methods have been applied [3] [25]. The latter build concept maps from handbooks about SCORM standards, based on hand-constructed patterns to match dependency parses, then use the concept maps to build ontologies. Our work also derives semantic knowledge from text, aimed at representing semantic relations among mathematics word problems.

Automated methods have also been used in construction of educational domain models for assessments [20], standards [9], and targeted prerequisites for learners [13]. Various approaches have been used to represent domain knowledge, including semantic networks with frames and production rules [23], or model-tracing architectures to identify problem-solving steps students take, including incorrect ones [2]. Model-tracing, inherently reactive, has been extended with tutorial actions to pro-actively guide students [12]. Other approaches to automatically generate metadata require existing domain ontologies [22]. Our goal is to develop a network of relations among problems that could be used proactively by ALEs or teachers to move students through the curriculum in a way that promotes optimal learning.

To represent mathematics word problems, we create a bag-of-words (BOW) vector for math words using methods similar to terminology identification [10]. In separate work, we use this vector to create similarity networks among problems [14]. A range of methods have been used to identify terms in product reviews [6], concepts in semi-structured data [4], technical language in patents [15], or domain-specific terminology in general [21]. Much of this work deals with identi-

Chap.	Sec.	Exer.	Text
2	1	19	The table shows a **proportional** relationship between x and y. Complete the table.
9	1	11	Solve the **inequality** x + 1 < 4. Then graph the solutions.

Figure 1: Sample word problems.

fication of multi-word noun-noun compounds of a technical nature, and ranking them. In contrast, the secondary school math terminology has few compounds, includes a mix of different parts of speech, and is non-technical. As in [21, 6, 4], we rely on relative frequency ratio [8] to distinguish the frequencies of words in our corpus from their frequencies in a large background corpus. Unlike most of this work, apart from [15], we developed annotation guidelines and measured interannotator agreement. We find an agreement of 0.81 among three annotators using Krippendorff's α (see below), compared to 0.76 (Fleiss's κ; a similar metric) in [15].

3. DATA: MATHEMATICS EXERCISES

The data consists of 3000 word problems from a Grade 7 mathematics curriculum. The problems, which can incorporate images, tables, and graphs, are instantiated through templates. Figure 1 shows two problem exercises from chapters 2 and 9, with words that evoke math concepts in boldface. Note that a template, $x\{+|-\}X\{< \mid >\}Y$, randomly generates instances such as $x+4>9$ or $x+1<4$. Depending on the number of instance variables and constraints, a template may generate a bounded or nearly limitless number of instances. In addition to the exercise itself, which may contain a few steps that are typically solved via multiple choice or fill-in-the-blank, learners are able to select a more detailed guided solution, or to view the steps to solve a sample problem instance. We created an XML parser to extract the text from the exercises, the guided solutions, and sample problems. The vocabulary analysis is based on the extracted text.

4. ANNOTATION AND RELIABILITY

At 4,495 words (not lemmatized), the curriculum's vocabulary is relatively small. Removal of typical stopwords leaves 4,283 words. An additional 103 words, while not typical stop words, have very high frequency across problems (e.g., *amount, answer, compare*) and are not likely to be useful for measuring semantic similarity among problems.

The terms we are interested in are those that are characteristic of the concepts the students should know to demonstrate mastery of the curriculum. The three co-authors, working independently, each labeled an initial sample of 100 words as math, non-math and other, based on initial guidelines. Because pairwise agreement can be high when a chance-adjusted agreement coefficient is low (the so-called paradox of kappa [11]), agreement was measured using both pairwise agreement and Krippendorff's Alpha [16], a metric that factors out chance agreement. Initially, pairwise agreement was 0.93, but Alpha was 0.54, which is rather low. The low chance-adjusted agreement was mainly due to inconsistency among annotators in looking at the contexts in which words were used, and also due to borderline cases. We wrote more explicit guidelines with examples (4 pages), then labeled two additional samples of 100 words each, computing agreement on each sample before proceeding to the next. On the second and third samples, pairwise agreement was 0.92 in both cases, and Alpha was 0.83 and 0.81. Given the high agreement and consistency across the second and third samples, we determined the labeling to be reliable. One of the co-authors labeled the remainder of the vocabulary, yielding 3832 words labeled as non-math, 571 as *math* and 92 as *other*. Only the words labeled as *math* and *non-math* were used to train the classifier.

1. Wolfram Mathworld
2. About.com: mathematics
3. Math domains in Google search results
4. Math domains in Bing search results
5. Digits math glossary
6. Relative frequency ratio

Figure 2: Features to represent vocabulary

5. CLASSIFICATION EXPERIMENTS

This section reports results from a suite of classification algorithms applied to the labeled data. To represent the vocabulary for the learner, we engineered features based on search and glossary information, and on a corpus-based metric. Two challenges for the classification were infrequency of the positive class (high data skew), and apparent non-linearity of the class separation. Of six learning algorithms, those that had best performance were most suited to these learning challenges, as described further below.

5.1 Feature Representation

We constructed a feature vector representation for the words with the 6 features listed in Figure 2. All feature values were scaled to be in the range of 0 to 1.

For the first two features listed in Figure 2, we used the functionality of Google Custom Search that permits customized searches to user-specified domains. For the first feature we queried mathworld.wolfram.com, and for the second we queried math.about.com. The value for each of these features consists of the total number of query returns, which can be arbitrarily large.

Google Custom Search can also be configured so that for the top ten returns to a query, each return consists of a triple with the url, a list of text snippets containing the term at that url, and the page title at that url. For the third feature listed in Figure 2, we query the web using this functionality, and calculate the feature value based on the triples for the top ten returns. Each time *math*, *mathematics*, or *arithmetic* occurs at least once in each element of a triple, a counter is incremented. The maximum value is thus 30.

Bing is a Microsoft search engine with an interface through which queries can be made programmatically. The interface returns the top 50 search results. Like Google searches, each result contains the relevant URL, snippets, and title of the page. As in the Google search feature, for the fourth feature in Figure 2, a counter was incremented whenever *math*,

Table 1: Classification Results

Classifier	Precision	Recall	Fscore	Sensitivity	Specificity	G-Mean
adaboost	0.89	0.90	0.89	0.42	0.97	0.64
bagging	**0.90**	**0.91**	**0.90**	0.41	**0.98**	0.63
rand-forest	**0.90**	**0.91**	**0.90**	0.45	0.97	0.66
SVM-poly	0.89	0.86	0.87	**0.68**	0.89	**0.78**
SVM-RBF	0.89	0.87	0.88	**0.68**	0.90	**0.79**
logistic regression	0.89	0.90	0.88	0.31	**0.98**	0.56

mathematics, or *arithmetic* occurred at least once in a triple element. Values are in [0,150].

The mathematics curriculum has an associated glossary of 246 math terms. It includes simple terms, e.g., "sphere," and compound terms, e.g., "associative property of multiplication." The glossary was expanded with the individual words in compound terms, excluding stop words. Thus for the compound term "associative property of multiplication", the words *associative*, *property* and *multiplication* were added. In this way, the glossary was expanded to 516 terms. A boolean feature value was used here to indicate exact occurrence of a word in the glossary.

Relative frequency ratio (RFR) measures relative frequency of a term in reference to a contrastive background corpus [6, 8]. The frequency of a word w_i in a corpus C, expressed as $FR(w_i, C)$, is its count normalized by size of the corpus. For a domain specific corpus, e.g., a mathematics text, the frequency of domain-specific terms should be higher than in a large, background corpus. The formula for RFR is:

$$RFR(w_i) = \frac{FR(w_i, DC)}{FR(w_i, BC)} \quad (1)$$

where DC is the domain corpus and BC is the background corpus. We tested RFR with two background corpora: the Open American National Corpus (OANC: N=22 $\times 10^6$) and English Gigaword, Fifth Edition (N=4,033 $\times 10^6$). Unsurprisingly, we found that the size of the background corpus is critical to the precision of the RFR measures. When we ranked Digits words by RFR scores using Gigaword, 306 of the words labeled as "math" occur in the top 1,000 words compared with 248 using OANC. Therefore we used Gigaword as the background corpus.

5.2 Classification

The labeled data was randomly split into a training set with 75% of the vocabulary (3301 terms) and a test set with 25% of the vocabulary (1101 terms). Using logistic regression, classification results yielded an overall precision of 0.87 and a recall of 0.88, compared with 0.78 precision and 0.25 recall for the *math* class. The low recall of math terms can be attributed to high class imbalance, where only 13% of terms are in the *math* class. Linear SVM also yielded poor results, suggesting that the classes cannot be linearly separated. To address the class imbalance, we use class weights for SVM, where we use polynomial and RBF kernels to address the non-linearity. Ensembles of weak learners also help with non-linearity. For each of three ensemble methods, *Boosting*, *Bagging* and *Random Forests*, we used 1000 *Decision Trees*.

Evaluation results are reported using precision, recall, f-measure, and g-mean [17]. The latter, the geometric mean of accuracy on the positive class (recall, or sensitivity) and accuracy on the negative class (specificity), is high when both accuracies are high and their difference is small. It is particularly useful when there are no criteria for constructing a cost matrix for errors in sensitivity versus specificity.

For the SVM classifiers, we used C=10,000. For the polynomial kernel, the degree was 4 and the class weights assigned to the math and non-math classes were 270 and 1350 respectively. For the SVM with the RBF Kernel, class weights were set to 200 and 1100.

6. RESULTS AND DISCUSSION

Table 1 shows the results for the six classification experiments. All the classifiers had high accuracy, due to the high class imbalance favoring non-math words. Accuracy on the math words (sensitivity), however, was relatively low for all but the SVM learners. The ensemble methods had higher precision on the math words ($\geq$ 0.78) but low sensitivity (0.41-0.46). The SVM learners had lower precision (about 0.5) and higher sensitivity (0.68). The logistic regression had very high precision on the math words (0.81) but very low sensitivity (0.32). For g-mean, all the classifiers had values above 0.50, indicating respectable peformance. The two SVM learners, however, had the highest g-means: 0.78 (polynomial kernel) and 0.79 (RBF kernel).

Manual error analysis of math words that were incorrectly classified by multiple learners indicated that many of the errors were due to polysemous words that have one or more non-math senses that occur with non-neglible frequency. This includes words like *point*, *dependent*, and *trial*. In WordNet [19], for example, *point* used as a noun has twenty-five senses, and fourteen senses used as a verb. Future work on the classification task will include investigation of features commonly used for coarse-grained word sense disambiguation, where accuracies of 88% have been achieved using lexical, syntactic and topical features [7] so that we can apply the same methods to new curricula.

7. CONCLUSIONS

The vocabulary classification task we address, to identify vocabulary that characterizes the semantics of a curriculum, differs from standard terminology detection, where the focus is on highly technical compound terms. It also differs from word sense disambiguation in that we are interested in binary classification of senses, based on the use of terms for a given curriculum. We have shown that human annotators can achieve very high pairwise and chance-adusted agreement. To avoid overfitting to a given curriculm, the features we used draw on external knowledge sources such as glossaries, web search and large background corpora. With relatively few such features and choice of an appropriate learning

algorithm, we achieve very high accuracy and good sensitivity, despite the small proportion of the positive class.

8. ACKNOWLEDGMENTS

We would like to thank the Research and Innovation Network at Pearson for support of this work.

9. REFERENCES

[1] V. Aleven, A. Ogan, O. Popescu, C. Torrey, and K. R. Koedinger. Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. Lester, R. M. Vicari, and F. Paraguaca, editors, *Intelligent Tutoring Systems: Seventh International Conference, ITS 2004*, pages 443–454. Springer, 2004.

[2] J. R. Anderson and R. Pelletier. A developmental system for model-tracing tutors. In L. Birnbaum, editor, *The International Conference on the Learning Sciences*, pages 1–8. Association for the Advancement of Computing in Education, Charlottesville, VA, 1991.

[3] L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, F. Wild, and others. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2):4–18, 2006.

[4] T. Atapattu, K. Falkner, and N. Falkner. Automated extraction of semantic concepts from semi-structured data: Supporting computer-based education through the analysis of lecture notes. In S. W. Liddle, K.-D. Schewe, A. M. Tjoa, and X. Zhou, editors, *Database and Expert Systems Applications*, number 7446 in Lecture Notes in Computer Science, pages 161–175. Springer Berlin Heidelberg, 2012.

[5] C. R. Beal, I. M. Arroyo, P. R. Cohen, and B. P. Woolf. Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions. *Journal of Interactive Online Learning*, 9(1):64–77, 2010.

[6] J. Broß and H. Ehrig. Terminology extraction approaches for product aspect detection in customer reviews. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 222–230, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[7] J. F. Cai, W. S. Lee, and Y. W. Teh. NUS-ML: Improving word sense disambiguation using topic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 07)*, pages 249–252, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[8] F. J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, 1993.

[9] H. Devaul, A. R. Diekema, and J. Ostwald. Computer-assisted assignment of educational standards using natural language processing. *Journal of the American Society for Information Science and Technology*, 62(2):395–405, 2011.

[10] P. Drouin, N. Grabar, T. Hamon, and K. Kageura, editors. *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, August 2014.

[11] A. R. Feinstein and D. V. Cicchetti. High agreement but low Kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.

[12] N. T. Heffernan, K. R. Koedinger, and L. Razzaq. Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18:153–178, 2008.

[13] S. Jain and J. Pareek. Automatic extraction of prerequisites and learning outcome from learning material. *International Journal of Metadata, Semantics and Ontologies*, 8(2):145–154, Jan. 2013.

[14] R. J. L. John, T. S. McTavish, and R. J. Passonneau. Semantic graphs for mathematics word problems based on mathematics terminology. In *WS-1: Graph-based Educational Data Mining (G-EDM 2015)*, 2015.

[15] A. Judea, H. Schütze, and S. Bruegmann. Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 290–300, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[16] K. Krippendorff. *Content analysis*. Sage Publications, Beverly Hills, CA, 1980.

[17] M. Kubat, Robert, and S. Matwin. When negative examples abound. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 146–153, London, UK, UK, 1997. Springer-Verlag.

[18] E. Melis, E. Andres, J. Budenbender, A. Frischauf, G. Goduadze, P. Libbrecht, M. Pollet, and C. Ullrich. Activemath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:385–407, 2001.

[19] G. A. Miller. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[20] R. J. Mislevy, J. T. Behrens, K. E. Dicerbo, and R. Levy. Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *JEDM - Journal of Educational Data Mining*, 4(1):11–48, Oct. 2012.

[21] Y. Park, R. J. Byrd, and B. K. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[22] D. Roy, S. Sarkar, and S. Ghose. Automatic extraction of pedagogic metadata from learning content. *Int. J. Artif. Intell. Ed.*, 18(2):97–118, Apr. 2008.

[23] S. Stankov, M. Rosić, Žitko, and A. Grubišić. TEx-Sys model for building intelligent tutoring systems. *Computers and Education*, 51:1017–1036, 2008.

[24] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[25] A. Zouaq and R. Nkambou. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1(1):49–62, Jan. 2008.

An analysis of peer-submitted and peer-reviewed answer rationales, in an asynchronous Peer Instruction based learning environment

Sameer Bhatnagar
Polytechnique Montreal

Michel Desmarais
Polytechnique Montreal

Chris Whittaker
Dawson College

Nathaniel Lasry
John Abbott College

Michael Dugdale
John Abbott College

Elizabeth S. Charles
Dawson College

ABSTRACT

This paper reports on an analyis of data from a novel *Peer Instruction* application, named DALITE. The Peer Instruction paradigm is well suited to take advantage of peer-input in web-based learning environments. DALITE implements an asynchronous instantiation of peer instruction: after submitting their answer to a multiple-choice question, students are asked to write a rationale for their choice. Then, they can compare their answer to other students' answers, and are asked to choose the best peer-submitted rationale among those displayed. We engaged in an analysis of student behaviour and learning outcomes in the DALITE learning environment. Specifically, we focus our investigation on the relationship between student proficiency, how students change their answers after reading each others' writings, and the peer-votes they earn in DALITE. Key results include i) peer-votes earned is a significant predictors of success in the course; ii) there are no significant differences between strong and weak students in how often they switch from the correct answer to a wrong answer after consulting peer-rationales, or vice versa; iii) even though males outscore females in conceptual physics questions, females earn as many votes from their peers as males do for the content they produce when justifying their answer choices.

Keywords

peer instruction, exploratory data analysis

1. INTRODUCTION

Active learning encompasses a broad movement in modern pedagogical practices, including any activities which engage the student as a part of the learning process, instead of passively receiving information during a traditional lecture. Such activities should encourage the student to read, write and discuss classroom content, as well as engage in higher order thinking tasks, such as synthesis and evaluation [1]. Active, cooperative, and collaborative learning practices have been shown to yield greater learning gains in science in engineering [8]. With the growing presence of on-line learning through instructional videos and accompanying readings, there is place for web-based activities which promote the same higher-order learning processes as those being used in more active classrooms.

This is where our research group found the need to develop the Distributed Active Learning Technology Integrated Environment (DALITE). The teacher-researchers in our group wanted a web-based homework system which would go beyond simply asking students for the answers to conceptual questions, by asking them to express the reasoning behind their thinking. This learning environment was meant to capture some of the higher-order thinking processes students engage in when reasoning about new concepts. DALITE is a system that would provide data on the mechanism of conceptual change, through the writings of students, as well as their evaluation of each other's work. What has emerged is an open source system which is being used in classrooms by learning science researchers who are also teachers.

Thus far, it has produced a dataset which can reveal new insights from the data on student production and consultation of answer rationales. Previous analysis of our work has already shown that students who use DALITE in college level physics classrooms do as well as those who use other on-line homework environments [2]. In the current study we analyze how the data on the production of rationales and the voting patterns can yield novel indicators of success and other characteristics of students.

This paper will begin with a description of the related field of Peer Instruction. The DALITE platform will then be described, as well as the most recent dataset collected. The focus of the analysis and results will be on the relationship between student proficiency, how students change their answers after reading each others' writings, and how many votes they earn for what they write. Finally we will discuss the potential and challenges that lie ahead, especially as student models are integrated into the DALITE system.

2. RELATED WORK

2.1 Peer Instruction

Peer instruction is a classroom practice popularized by Eric Mazur of Harvard University [3]. In its most common instantiation, the classroom script goes as follows:

1. The teacher displays a multiple choice question to the whole class, and asks everyone to reflect, and individually choose what they think is the correct answer. This is typically done by giving each student a handheld clicker, which transmits the answer to a receiver plugged into the teacher's computer.
2. The teacher displays a bar chart showing the distributions of answer choices for the whole class. The students are then prompted to discuss their answer choice with their peers for several minutes, after which they are given the opportunity to answer the question again using their clicker.
3. The teacher shows the new distribution of answers. Typically, after the peer discussion, there is a major shift towards the correct answer.

Making this a regular practice in class has been shown to yield higher learning gains [7] and lower dropout rates [4] compared to conventional, teacher-centered, lecture style courses. However it is very difficult to capture what is actually happening during the student discussions. What is actually being said to convince someone to change their answer (or at least change their rationale for their answer choice)? How does that relate to cognitive theories of learning? DALITE collects information exchanged in written form through Peer Instruction features embedded within a web based learning environment, namely answer rationales and votes. The information hereby collected allows us to better address the above questions empirically.

3. THE DALITE PLATFORM

DALITE is a web-based drill and practice platform that contains introductory level physics problems. It has an interface for the student to work on physics problems, and a teacher interface to manage the learning content.

3.1 Student interface

Students log into DALITE, and work on an assignment which typically contains four to six multiple choice questions. For each question, there are three screens they must flip through, each with the following structure:

1. The question is displayed, and the student selects one of the multiple choice answers. They are then prompted to write a couple of sentences that explain why they selected their answer choice. These little paragraphs will from now on be referred to as "rationales".
2. Once a rationale is given, the system presents two columns: one for their answer choice, and one for another choice to the question. Each column contains four rationales, written by previous students. The aim is to give students a chance to reflect on their thinking by providing them with an opportunity to compare and contrast other rationales and change their mind. The student is prompted to read the rationales from the two columns, and decide whether they would like to keep their choice, or switch. What's more, the student is asked to choose one rationale out of the ones displayed that they best like. They can also simply cast an "empty ballot", in effect saying that none of the other students' rationales were convincing. This up-voting process is anonymous.
3. The third screen recaps everything that just happened: the question is shown, alongside their two answer choices (one from each of the previous two screens). What's more, the rationale they originally wrote is reflected back to them, right next to a rationale written by an expert for the correct answer.

3.2 Teacher Interface

When teachers login to the system, they can:

- upload new questions to the database. This requires that the question be of multiple choice format. The teacher must specify the correct answer, with a rationale justifying that answer choice. The teacher must also identify a "second best answer", which would be used for the second column of the second screen (described above) should the student answer correctly on their first attempt. Teachers can also add "tags" to the question, which describe the content of the question.
- build new assignments based on questions already in the system.
- observe the results of assignments done by their students. The current reporting tool gives the teacher a mini grade-book for each assignment, where each student is a row, and each question is described by two columns: one for the student's first answer, and one for their second answer. Teachers can quickly get a sense of where the students are getting confused, as cells are coded green for the correct answer, and red for the incorrect answer. Transitions from red to green are signs that the rationales in the database are doing their job of convincing students to move away from the wrong answer, while transitions from green to red show that the students' conceptual understanding is shallow.

4. THE DATASET

Although DALITE has been in use for the last five years, it was during the Fall semester of 2013 that a comprehensive dataset was collected in a systematic manner over the entire term. The cohort was comprised of 144 students, spread out in five groups, taught by four different teachers, across three colleges. The system was used to teach freshman year, calculus -based Newtonian Mechanics. This is at a level equivalent to grade 12 in high school in the US and other Canadian provinces.

4.1 Data from within DALITE

Over the course of the semester, 80 question items were assigned by the different teachers, 40 of which were completed by at least half of the entire cohort, providing data on over 7000 student-item pairs.

Each student-item pair in the dataset includes the initial answer, the rationale, and the final answer. A separate table

in the database keeps a count of how many peer-votes are earned by any given rationale.

4.2 Data from classrooms

For each student in the five experimental groups, as well as one control group (which did not use DALITE), the following data was collected inside their classrooms over the course of the semester:

Pre-Post FCI The Force Concept Inventory (FCI)[5], is a questionnaire of 30 conceptual questions about the Newtonian concept of force. The exact same questionnaire was administered on the first day of class, and then again on the last day of class, for each of the groups, in order to compare the learning gain between the DALITE users and students who did not use DALITE. The item-by-item results of this questionnaire can be compared to a FCI dataset which holds the results of more than 13000 students from across Canada and the U.S.

Midterm & Final Exam Grades The Newtonian Mechanics course commonly has three major themes: Kinematics, Dynamics, and Laws of Conservation. This lines up with the three midterms for which each student's grade is recorded. Finally, for each student, the final exam grade is broken down by the result on the multiple choice section (typically more conceptual questions, and hence more similar to DALITE), and the long-answer section (typically computations and problem-solving).

5. RESULTS

During the Fall 2013 study, four experimental groups were assigned DALITE specifically as homework for their students. Following are the key results:

Student Success How well students succeeded on DALITE questions had 0.50 and 0.60 correlations with their performance on the conceptual, multiple choice part of their final exam, and the post-semester FCI questionnaire, respectively. This provides some measure of the reliability of this relatively new homework system.

Also a linear model was fit to predict a student's final grade based on statistics from their DALITE account. The fraction of questions students answered correctly out of those they attempted, as well as the total number of votes they accumulated, were both significant predictors of their final grade in the course ($R^2 = 0.24$, $p<0.001$). This predictive power of DALITE emerges as early as after the first third of the course, meaning the teacher can get early indicators of which students are at risk for the midterm.

In a related line of questioning, the data was partitioned by gender of the students. Male students did significantly better than female counterparts in all measures of conceptual understanding from the classroom (pre-term FCI score, pre-post term gain on FCI, conceptual questions on final exam). This is in line with previous work looking into the gender gap in introductory physics [6]. This gap was found in the DALITE data as well, with males getting 20% more of the questions items right ($p<0.001$).

Patterns in how students change their answer choices Over the course of the semester, students who started with the right answer, only switched to the wrong one 1 out of 10 times. However, when they started with the wrong answer, they switched to the correct answer 3 out of 10 times after reading their peers' rationales. This gives some measure of overall quality of the rationales currently in the database: the rationales to the wrong answers are not highly persuasive, and there are at least some rationales for the correct answers which can convince students to change their minds when they are wrong.

Factors affecting answer change When the data was separated into quartiles for the final course grade, it was found that strong students were as likely as weaker students to switch from the right answer to the wrong answer. In addition, the converse was also true: weaker students were as capable of switching to the right answer when they got it wrong on their first attempt. There was some effect herein due to the teacher: the experimental groups that regularly discussed DALITE homework in class, were significantly more likely to change their answer when in DALITE. In the group that used DALITE purely as extra homework, answer switches were much less likely ($p<0.001$).This may indicate that the students who are reminded that the system is a valuable tool, are more engaged with the system, and take the time to more carefully read each others' rationales.

The well known gender gap mentioned, males outscoring females in conceptual physics questions, interestingly disappears if we measure correctness based on the second attempt: female students choose the wrong answer 20% more often on their first attempt, but after reading peer-written rationales, they identify the correct choice just as often as males.

Who amasses more peer votes? Students from the stronger half of the cohort earned, on average, more than two times as many votes as those from the bottom half. What's surprising is that this pattern holds true for the wrong answers as well: even when the strong students are wrong, they are twice as convincing as their weaker peers. This is especially relevant in light of the fact that 1/3 of all the votes cast over the term were for rationales to wrong answer choices. In parallel to this finding, when we looked only at rationales justifying the correct answer choice, it was found that weak students earned as many votes as their stronger colleagues. This seems to indicate that even if a student did not perform as well on tests, when they were right on a particular conceptual question, they were able to justify their understanding as well as stronger students.

The gender gap discussed earlier, was also lost when looking specifically at the voting data. Even though males achieve higher grades on conceptual questions, females of all strengths earn as many votes for their rationales as the males. This tends to indicate that females produce content justifying their understanding

that is as valued by their peers as rationales written by males.

6. DISCUSSION

The key results described above show the potential for DALITE to be an effective tool for teachers to probe their students' deeper understanding of concepts in physics, and identify students at risk of failing midterms and final exams. The data on how students change their answers based on the writings of their peers, and which rationales they vote for, may give teachers and researchers insight on what words can trigger conceptual change in different types of students. Finally, the data shows that students who may not perform as well on summative evaluations, are still able to produce valuable content when justifying their understanding.

7. FUTURE WORK

Future directions of research on this project include capturing not just which rationales got voted for, but who is casting the votes, and in what context. The goal is to explore what features in student written text have an impact on changing peer conceptions of scientific concepts. Do students learn from stronger students, or only those within their Vygotskian zone of proximal development [10].

Another important direction would include collaborative filtering techniques, which are traditionally applied to recommender systems, such as in the e-commerce setting, where a users-by-item ratings matrix is used to predict what items new users would most likely enjoy. Recently such techniques have been applied in the context of educational data mining, where the matrix is now student-by-item performance, and factorization leads to estimates of the probability of another student getting a new item correct [9]. With the ratings data collected, the system may be able to deliver individualized rationales to different learners with the same misconceptions to the same question item. What is most promising is how this open-source tool creates a venue for learning science researchers to ask questions regarding higher-order learning processes, such as evaluation and synthesis, and for the EDM community to test-drive different text mining techniques in a real classroom setting.

8. ACKNOWLEDGMENTS

The strength of the DALITE platform resides in the database of student rationales, so the students who have used this platform for learning must be thanked for providing this rich set of data. This work has been funded through the Programme d'Aide à la Recherche sur l'Éducation et l'Apprentissage(PARÉA), administered by the Ministère d'Éducation et Loisirs de Quebec.

9. ADDITIONAL AUTHORS

Kevin Lenton, Vanier College

10. REFERENCES

[1] C. C. Bonwell and J. A. Eison. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991.

[2] E. Charles-Woods, C. Whittaker, M. Dugdale, N. Lasry, K. Lenton, and S. Bhatnagar. Beyond and within classroom walls: Designing principled pedagogical tools for students and faculty uptake. In *Computer Supported Collaborative Learning (in press)*, 2015.

[3] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, 2001.

[4] A. P. Fagen, C. H. Crouch, and E. Mazur. Peer instruction: Results from a range of classrooms. *The Physics Teacher*, 40(4):206–209, 2002.

[5] D. Hestenes, M. Wells, and G. Swackhamer. Force concept inventory. *The physics teacher*, 30(3):141–158, 1992.

[6] L. E. Kost, S. J. Pollock, and N. D. Finkelstein. Characterizing the gender gap in introductory physics. *Physical Review Special Topics-Physics Education Research*, 5(1):010101, 2009.

[7] N. Lasry, E. Mazur, and J. Watkins. Peer instruction: From harvard to the two-year college. *American Journal of Physics*, 76(11):1066–1069, 2008.

[8] M. Prince. Does active learning work? a review of the research. *Journal of engineering education*, 93(3):223–231, 2004.

[9] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme. Factorization techniques for predicting student performance. *Educational Recommender Systems and Technologies: Practices and Challenges*, pages 129–153, 2011.

[10] L. Vygotsky. Interaction between learning and development. *Readings on the development of children*, pages 34–41, 1978.

Learning Analytics Platform, towards an open scalable streaming solution for education

Nicholas Lewkow
McGraw-Hill Education
281 Summer St.
Boston, MA
nicholas.lewkow
@mheducation.com

Neil Zimmerman
McGraw-Hill Education
281 Summer St.
Boston, MA
neil.zimmerman
@mheducation.com

Mark Riedesel
McGraw-Hill Education
281 Summer St.
Boston, MA
mark.riedesel
@mheducation.com

Alfred Essa
McGraw-Hill Education
281 Summer St.
Boston, MA
alfred.essa
@mheducation.com

ABSTRACT

Next generation digital learning environments require delivering *just-in-time feedback* to learners and those who support them. Unlike traditional business intelligence environments, streaming data requires resilient infrastructure that can move data at scale from heterogeneous data sources, process the data quickly for use across several data pipelines, and serve the data to a variety of applications. As a solution to this problem, we have designed and deployed into production the Learning Analytics Platform (LAP), which can ingest data from different education systems using standardized IMS Caliper events. The education events are triggered by student and instructor activity within Caliper instrumented learning systems. Once sent to the LAP, events are transformed and stored in a data store where they can be used for student, educator, and administrator visualizations as well as education driven analytics research. Two McGraw-Hill Education platforms, Connect, used for higher education, and Engrade, for K-12, are currently instrumented to send the LAP event data which in turn feeds visualizations for educational insight. Future plans for the LAP include collection of education event data from a wide variety of proprietary and open source education platforms, computational engines for predictive analytics, and an open API for third-party analytics using LAP data.

Keywords

Learning Analytics, Event Processing, Heterogeneous Data, Streaming Data, Parallel Architecture

1. INTRODUCTION

It is the goal of next generation digital learning systems to use big data and analytics to advance learning outcomes. These next generation systems should be able to provide just-in-time feedback to students and educators with an aim to increase the efficiency and effectiveness of digital education. Further, with the increasing instrumentation of all digital media, digital learning environments should be instrumented in a way that allows important education data to be collected in a standardized fashion for both real-time and after-the-fact (batched) data analysis. This task requires large scale processing of streaming data utilizing massively parallel architectures which may ingest, store, and analyze data in real-time.

Our solution to this problem is the Learning Analytics Platform (LAP). The LAP is designed to ingest educational data from present and future education platforms in the form of standardized events using the IMS Caliper spec [2]. Two existing education platforms, Connect for higher education and Engrade for K-12, have already been instrumented to create and ship Caliper education events to the LAP. Once ingested by the LAP, the data is transformed and stored for building of visualizations used for educational reports [4]. These 'insights' include several real-time statistics for students and educators including time-spent, outcomes, submission times near due dates, attendance, and class performance comparisons. Additionally, messages indicating negative trends, such as repeatedly starting assignments near or after the due date, are also presented to the user.

To meet the requirement of providing just-in-time feedback to students and educators, an analytics platform must also be have a parallel architecture which may effectively ingest streaming data. There are several proposed requirements for a streaming data architecture, including the ability to handle data imperfections, generate predictable outcomes, and to guarantee data safety and availability [5]. Additionally, the architecture should be automatically scalable and fault tolerant for both software and hardware failures, particularly, it must not lose any event data under any circum-

stance. Our architecture has the additional requirements of ingesting data from heterogeneous sources and performing data transformations using different data pipelines. The LAP fulfills all of the above requirements in its current version with further refinements and additions planned for the near future.

Details about the LAP are discussed in the following sections including information about the standardized data format which was used, IMS Caliper events, as well as specifics on the LAP architecture and performance. Information regarding future versions of the LAP is also discussed followed by concluding remarks.

2. STANDARDIZED CALIPER EVENTS

With the continual adoption of digital education systems a global standard for educational event data, which is generated from a large diversity of heterogeneous systems, has become increasingly sought after. While there has been advancement in this area by the Tin Can API [1], the IMS Global Learning Consortium has proposed a schema-driven solution to this problem with their Caliper event spec [2]. JSON-LD (linked data) is used for the Caliper events as a way to link specific, normalized fields within a set of events [3]. Using the Caliper events, data from heterogeneous learning systems can be created, transmitted, and collected for analysis in a global and standardized fashion.

Caliper events strive to create a generalized framework that can be utilized by all types of learning events ranging from a student using an interactive education tool, such as a learning game, to an educator recording attendance in their class. The Caliper events are based on the data triple of **"Actor" - "Verb" - "Object"**. As an example, the event for a student submitting an assessment (homework, quiz, test, etc.) would have the **Actor** be the student, the **Verb** be the submission of an assessment, and the **Object** would contain information on the assessment being submitted, potentially how long it took to complete.

In order to utilize the Caliper event spec, learning platforms must be instrumented to create events when actions occur by either students or educators. Currently the Connect and Engrade systems are instrumented to to create Caliper events when actions occur, and to send the events to the LAP. Instrumentation is unique to the system in question and greatly depends on how that system's data is stored. In the case of Connect, Caliper events are created through a series of database triggers when actions are taken by students. The database triggers are automated to create the Caliper events using tables from their system databases when new information is passed to the system from a user. Future plans include instrumentation of several new external systems, allowing for increasingly rich data in the LAP for analysis and visualization.

3. ARCHITECTURE

Increasing instrumentation of sensors and digital media requires streaming analytics architectures to analyze data in real-time. Attention was paid to the development and design of the LAP architecture to ensure that it met all requirements of a parallel streaming system containing both a data store and an analytics engine. Key features include auto-scaling fundamental architecture components with varying load, fault tolerance for both hardware and software failures, and the ability to process data and have it available for output API access immediately.

In the simplest of descriptions, the LAP is designed to:

1. Receive learning events from external applications through an ingestion API.
2. Send raw events directly to long-term storage.
3. Validate each event for expected fields and types.
4. Process the events, which requires application dependent data transformations.
5. Store those events in the data store according to application dependent schema.
6. Query the data store and perform transformations and aggregations as needed for the output API.

Figure 1 displays a high level view of the LAP architecture with learning events from three separate external applications coming into the LAP. Once the events are received by the LAP they are transformed according to an application specific schema and stored in the data store. When a user requests an insight visualization, the LAP output API is called by the external application. This triggers the results and analytics service to query the data store, aggregate and transform the data as needed, and pass it to the insight service where the data is used to build the appropriate insight visualization which is then passed to the user.

The ingestion API and collection service are configured to receive IMS Caliper events sent from external applications through sensors which have to be implemented in those external applications. To maximize performance over high-overhead HTTP, several events are sent simultaneously in an event container. The number of events sent in a single container can range from one to tens of thousands. Once a container of events is received it is immediately sent to a long term storage system for backup purposes. The container is then opened and the events within are validated, transformed into an application specific schema, and sent to the data store.

The data store utilizes the non-relational database MongoDB, which allows for great flexibility in data model schema. Events from external applications are not guaranteed to come into the LAP in chronological order so the data model schemas were developed to create a deterministic data store state from events coming in arbitrary order. The data model employed for the current two applications consists of a two schema model; one document type holds the student level data and the other holds the class level data. Building of the insight visualizations then requires the results service querying N student documents for a class with N students, and 1 document per class for a total of $N + 1$ documents.

The insight service is built external to the LAP to provide the user with the requested analytical visualization. Once the data store is queried and the appropriated documents are

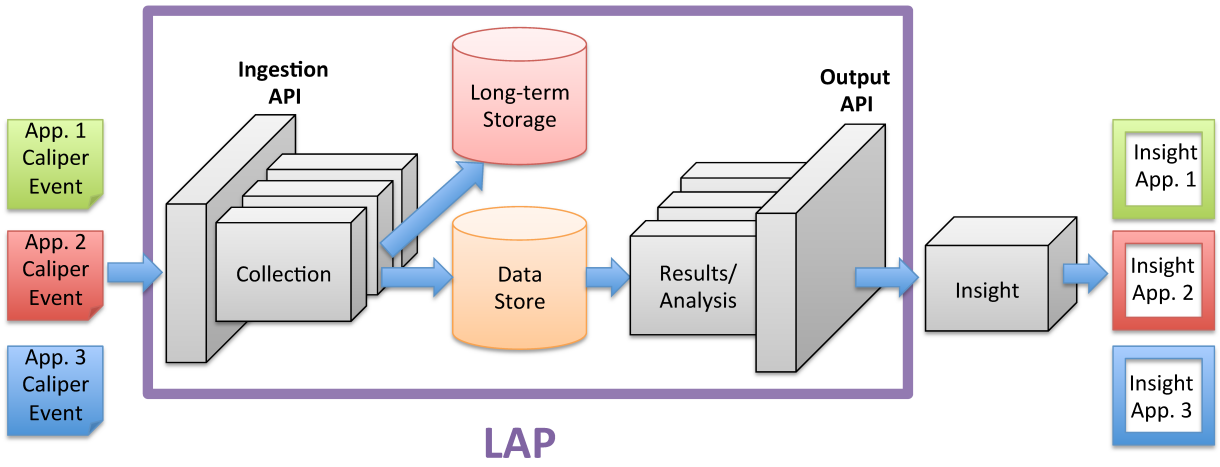


Figure 1: High level view of the LAP architecture. Learning events are passed into the LAP through the ingestion API and visualizations 'insights' are produced through the insight service calling the LAP's output API. Internally, the LAP consists of a collection service, a data store, long-term storage, and a results service, which also performs analytics. Several instances of the collection and results services run in parallel.

returned, they are aggregated, transformed, and returned to the output API to provide the client with data to build the visualization. This process is very lightweight and fast, allowing for quick feedback and response to the user who has just finished an assessment.

The technology used for the LAP was chosen to be lightweight, reliable, and adaptable for future system iterations. The collection, results, and insight services were all built using Node.js which allows for asynchronous communication run in parallel across several service instances. The analytical visualizations were built using JavaScript with AngularJS and D3.js frameworks which provide fast, responsive, and interactive interfaces that are customizable. The data store was built using MongoDB with a three member replica set, allowing for highly available data and low latency access for both read and write operations. Amazon Web Services (AWS) were used to host the LAP utilizing load balancing technologies routed to EC2 instances. An AWS S3 bucket was used for long-term storage of the raw event containers. Details pertaining to the performance of the LAP as well as future plans for the system follow.

4. PERFORMANCE

Performance is very important to the success of the LAP. It is imperative that the LAP be able to ingest data from several external systems during their peak times simultaneously in a manner which does not delay the real-time analysis on the output of the system.

In testing the performance of the LAP, two main points within the system were identified as the potential bottlenecks. The first of these points is the ingestion of events from external applications into the LAP while the second is the querying, aggregation, and analysis done by the results service prior to returning data to the output API.

For the current two external systems sending data to the LAP, our anticipated peak load is around 0.1 MB/sec. While this load is not particularly large, future plans include ingesting events from many more external systems, so the desired performance should easily be at least ten times larger at around 1 MB/sec. To test the performance from event reception to data store insertion, an automated script was developed which creates 10 threads with each sending a series of containers with varying numbers of events to the LAP running three instances of the collection service. The processing time, from data being sent from an external application to be inserted into the data store, was then measured as a function of number of events per container. Figure 2 displays the results of this test with collection rates as MB/sec and the number of events per container ranging from 1 to 1000. The results shown in Figure 2 are informative for a few reasons. First, it is clear that high collection rates into the system requires more than one event to be sent per container. In particular, the LAP can process 0.1 MB/sec or more if the events are sent with at least 4 per container. To reach our desired bandwidth of ten times our current peak, 1 MB/sec, requires sending about 75 events per container or more with three collection instances. The second realization from Figure 2 is that the collection rate of the system somewhat flattens out between 500 and 1000 events per container, making it less efficient to process these larger containers. This effect ultimately has to be weighed against the network bandwidths in sending events from external applications and has not yet been tested. It should also be mentioned again that these performance tests were done with three instances of the collection service. Increased rates could always be achieved by increasing the number of collection service instances, but these tests were done to determine collection rates for a static number of instances.

The second potential bottleneck in the LAP is the querying of the results service and the load on the output API. Currently the LAP is in a trial mode with tens of thousands of users. For this relatively low number of users the load on the output API is not a major concern and detailed performance testing has not yet been done. The implementation

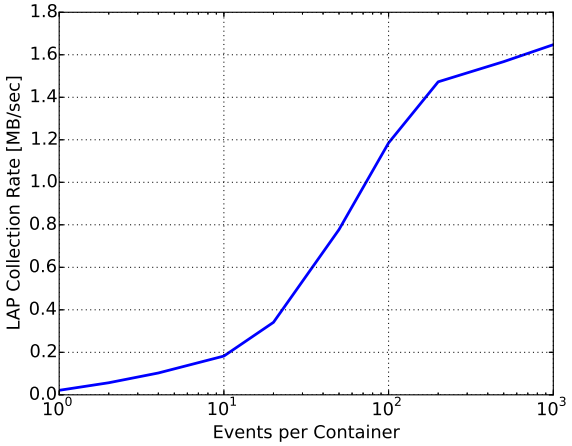


Figure 2: The collection rate of Caliper events, in [MB/sec], processed by the LAP is shown as a function of number of events per container. 10 parallel threads were used to send data to the LAP during this test, simulating the load from several external systems sending events simultaneously. The system tested had 3 instances of the collection service operating in parallel.

of auto-scaling within AWS, however, should easily handle large spikes in front end usage during peak hours.

5. FUTURE VERSIONS

The current LAP is the first iteration of a production system. It supports two external education systems, can support more than ten times the anticipated peak load, and has a modest analytics layer within its architecture. The next version of the LAP is currently being developed with the goal of supporting many more external systems with totals of tens of millions of users. In addition to increased user load, the future plans for the LAP include a substantial computational layer, opening up the possibility for richer analytics, as well as an open API for third party analytics to be done using LAP data.

The initial success of the current version of the LAP has led to plans for instrumentation of several new educational systems so their data may be ingested by the LAP. To be able to handle the increased numbers of users with the LAP, several changes and additions are needed. The most drastic of these changes is switching to a completely AWS system, fully utilizing Amazon cloud technologies [6]. Moving the LAP to a full AWS stack will allow for massive scalability and the ability to store data, perform analytics, and give support to millions of students, educators, and researchers. Further, future versions of the LAP will also have the ability to perform more advanced analytics including predictive analytics, machine learning algorithms, and large distributed calculations and aggregations. Incorporating a distributed calculation layer into the LAP will allow for a richer set of analytics to be performed and thus give the ability for deeper insight into large educational data sets.

Implementation of an open API for data consumption and analytics by third parties is also planned for the LAP. One of the intended features of the LAP is the lack of PII data held within the data store. The de-identified of the LAP data allows for third parties to ingest and do analysis on our data without concern for privacy. Creating an open API for the LAP will help push the fields of learning analytics and educational science by allowing researchers greater access to student and educator data.

6. CONCLUSIONS

We have built a platform able to ingest, store, and analyze data from external learning applications in a scalable fashion. Two existing applications, Connect and Engrade have been instrumented to create and send standardized Caliper learning events to the LAP. Once received, the learning events are transformed within customized data pipelines and stored within a fast data store, implemented using non-relational MongoDB. Analysis is done on the stored learning events, creating visualizations of educational insight for students and educators. The architecture of the LAP allows for just-in-time feedback with the insight visualizations to its users. The current version of the LAP is able to process and store education events ten times faster than is required for peak usage by the current two applications interfaced with the LAP. Future versions are planed for the LAP and will include a complete backend stack which is hosted by AWS and able to auto-scale across the entire platform. Additionally, an advanced computational layer and open API are planed for future versions of the LAP. It is our vision that present and future iterations of the LAP may provide the analysis, high quality educational data, and predictive analytics to drive the next generation of education.

7. ACKNOWLEDGMENTS

This paper is based on work supported by the McGraw-Hill Education Digital Platform Group (MHE DPG). We would like to extend our appreciation for all the help and the informational support provided by the Analytics team at DGP and the MHE CDO Stephen Laster. Despite provided support, any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

8. REFERENCES

[1] Experience API Version 1.0.0. Technical report, Advanced Distributed Learning, 2013.
[2] Learning Measurement for Analytics. Technical report, IMS GLOBAL Learning Consortium, 2013.
[3] JSON for Linking Data, 2015. http://json-ld.org/.
[4] J. Feild. Improving student performance using nudge analytics. In *Educational Data Mining*, 2015.
[5] M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. *SIGMOD Record*, 34(4), 2005.
[6] E. Swensson. Big Data Analytics Options on AWS. 2014.

Improving Student Performance Using Nudge Analytics

Jacqueline Feild
McGraw-Hill Education
281 Summer Street
Boston, MA USA
jacqueline.feild@mheducation.com

ABSTRACT

Providing students with continuous and personalized feedback on their performance is an important part of encouraging self regulated learning. As part of our higher education platform, we built a set of data visualizations to provide feedback to students on their assignment performance. These visualizations give students information about how they are doing compared to the rest of the class, and allow them to compare the time they spent on assignments across their courses. Included in the feedback are 'nudges' which provide guidance on how students might improve their performance by adjusting when they start or submit assignments. In order to understand what nudges to provide to students, we analyzed historical data from over 1.4 million students on over 27 million assignment submissions to find student performance trends. The data confirmed that student performance significantly decreases when assignments are started on the same day they are due and when they are submitted after the due date. We used these findings and the past and current performance of each student to display nudges relevant for them in their visualizations, highlighting actionable strategies for improving future performance.

Keywords

self-regulated learning; data visualization; data mining

1. INTRODUCTION

Self-regulation is a trait very often associated with highly effective learners [6, 1]. Feedback is an important part of the process of self-regulation, as it allows students to evaluate their performance, to decide what actions might improve their future performance and to make adjustments to their learning processes [3, 4]. Feedback can be provided in a variety of ways, but it is especially effective when it is personalized and given in near real-time. In this paper, we describe a set of data visualizations we incorporated into our higher education platform, Connect, to provide students with exactly this kind of continuous, easy to understand feedback on their assignments to encourage the development of self-regulated learning.

Specifically, these visualizations allow students to see how they are doing on assignments as soon as they are graded. In two easy-to-understand visuals they can see trends in their performance over the semester, compare their performance to the rest of the class, and compare the time they spent on each assignment across courses. In addition to this information, we use 'nudge analytics' to provide personalized messages to encourage students toward actions that might improve future performance based on patterns in historical data [2, 5]. The word 'nudge' means to encourage someone to do something, and nudge messages are an unobtrusive way to push students toward better behavior, while leaving the choice to change up to them.

To find relevant nudges, we performed exploratory data analysis on eight months of student submissions to our higher education platform, including over 1.4 million unique students and over 27 million assignments. Our goal was to find trends in the data that identify factors that lead to decreased performance for most students. In this paper, we explore the assignment submission trends by day of semester, day of week, hour of day, and started and submitted time.

2. CONNECT INSIGHT FOR STUDENTS

McGraw-Hill Education offers a teaching and learning environment, called 'Connect', for higher education. This environment allows instructors and students to manage assessments, and access ebooks and other instructional materials. As part of Connect, we built a set of visualizations called 'Insight' to help students understand their performance on assignments. These visualizations provide important feedback to students as soon as assignments are graded in a way that is easy to understand. The interactive nature allows students to make decisions about what actions might improve their future performance.

The example visualization in Figure 1 answers the question, 'How am I progressing?' and shows a student their scores on assignments in a particular class over time. The yellow trend line shows the students scores and the blue trend line shows the class average on assignments. Clicking each data point opens a right-hand panel with more details, including the nudges toward better performance if applicable. In the following sections, we will describe our analysis for determining these messages in more detail.

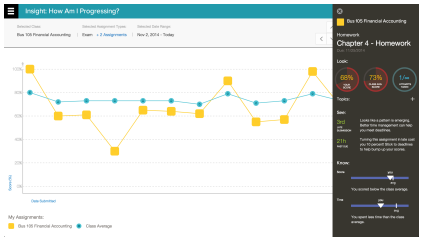


Figure 1: A visualization to answer the question 'How am I progressing?'

3. EXPLORATORY DATA ANALYSIS

In this section, we describe our analysis of eight months of historical data from assignment submissions. The goal of this analysis is to find trends in student behavior that negatively influence performance. This will help us identify the nudges that are supported by the data, and can be used to encourage students towards performance increasing behaviors.

We used historical data collected by Connect during the spring and summer semesters of 2014. This included data for 80,000 class sections taught by 29,000 instructors to 1,400,000 students. The result is over 27 million assignment submissions.

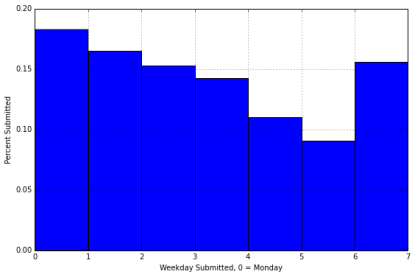
The data we used for our analysis was given to us by the Connect team from their database designed for end users, and it was not optimized for analytics. Instead, we used the existing fields for assignment submissions, including the assignment type (homework, quiz, exam, etc), start date, completion date, due date and outcome. From this data we computed a number of derived fields, including the hour of the day, day of the week and, day of the semester an assignment was submitted. We also computed the number of minutes before the due date each assignment was started and submitted.

Given these attributes, we focused our analysis on trends in assignment started and assignment submitted times. In the following sections we explore the assignment submission trends by day of semester, day of week, hour of day, and started and submitted time.

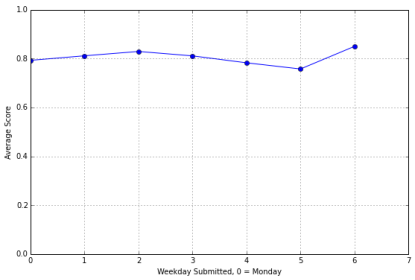
3.1 Day of the Semester

First we asked the question, does performance decrease during the semester? To start, we looked at the percent of assignments submitted on each day in our data set. This shows an interesting repeating pattern of the highest number of submissions on Monday and the lowest number of submissions on Saturday. It also shows a drop in submission volume in the middle of the spring semester, which can likely be explained by the week long spring break that occurs during this time period. Other than this decrease, submission volume remains consistent over both the spring semester and summer semester.

In order to understand student performance, we looked at the average score for assignments submitted on each day in our data set. We see a trend of decreasing performance toward the end of the spring semester (starting just before day 100). We see a similar downward trend for scores toward the end of the summer semester as well.



(a) A histogram of the percent of assignments submitted on each day of the week



(b) A plot of the average score for all submissions for each day of the week.

Figure 2

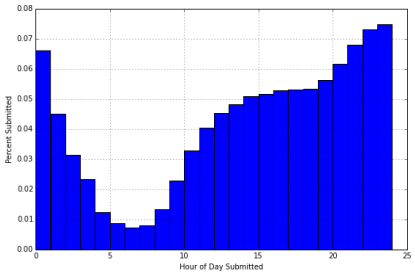
Unfortunately, there is not a clear delineation between the spring semester and the summer semester, and between the summer semester and the following fall semester, as different schools schedule classes over different time periods. Information on when classes start and end is not included in our data set, so further research is needed to confirm that this trend exists on a normalized data set.

3.2 Day of the Week

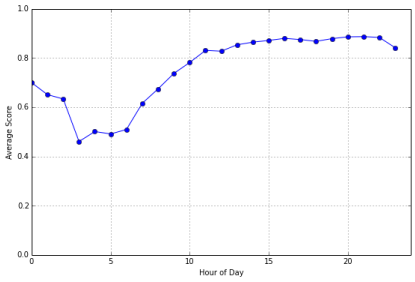
The previous analysis showed that performance decreased toward the end of the semester, but we also want to know, does performance decrease on any day of the week? Figure 2a shows the percent of assignments submitted on each day of the week. This confirms what we saw in the previous section, that the most number of assignments are submitted on Monday, while the least number of assignments are submitted on Saturday. Figure 2b shows the average score for assignments submitted on each day of the week. As expected, this shows that there is no performance advantage to submitting on a particular day of the week.

3.3 Hour of the Day

Following this analysis of scores over the semester and week, the obvious next question to explore was, does performance decrease when assignments are submitted at particular times of the day? Figure 3a shows the percent of assignment submitted during each hour of the day. This shows that most assignments are submitted between 12pm and 12am, with an increase around 8pm. While submissions do decrease in the early morning hours, there are still many submissions between 12am and 8am.



(a) A histogram of the percent of assignments submitted during each hour of the day



(b) A plot of the average score for all submissions during each hour of the day

Figure 3

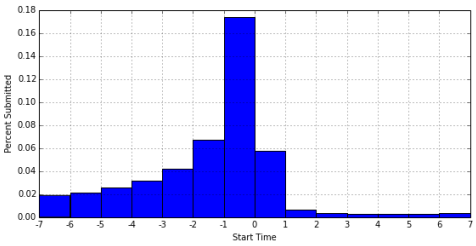
Figure 3b shows the average score for assignments submitted during each hour of the day. The average score is similar between 10am and 11pm, but steadily decreases from 11pm to 6am before increasing again. The decrease in score is significant, going from an average score of 89 at the peak hour to an average score of 46 at the lowest hour.

Unfortunately, this data represents students in many different timezones, but the the date fields are all represented in Eastern local time, where the platform servers are located. This means that we cannot draw the conclusion that submitting in the early morning hours leads to lower scores from these plots. Further work is required to obtain student time zone information and to clean the data by adjusting dates to each students local time.

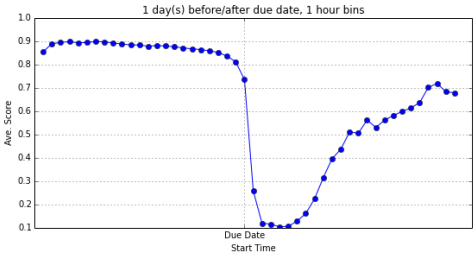
3.4 Start and Submit Time

We also looked at when a student started and submitted assignments in relation to the due date to answer the question, does assignment start time or submission time affect performance? Figure 4a is a histogram showing the percent of assignments started each day before and after the due date. The zero on the x-axis represents the deadline, so the bar between -1 and 0 represents all of the assignments that were started the same day they were due. The interesting trend in this plot is that most late assignments are started the day after the due date. This means that most late assignment could be avoided without drastic behavioral changes.

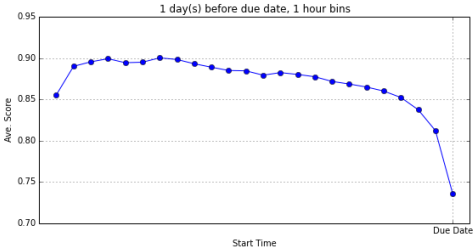
Figure 4b shows the average grade for assignments started at different points before and after the due date. The due date is in the center, and each data point to the left and right represents a 1-hour range. So the data point at the due date represents the average score of all of the assignments started



(a) A histogram of the percent of assignments started each day before and after the due date. The zero on the x-axis represents the deadline.



(b) A zoomed in version of (b) where each data point represents a 1-hour range.



(c) A zoomed in version of the left half of (c)

Figure 4

within the last hour before the due date. The point just to the left of the due date represents all of the assignments started between 1 and 2 hours ahead of the due date, and so on. In total, the plot shows one day before and after the due date. This shows that there is a decrease in average score as assignments are started closer to the deadline.

For a more detailed view, Figure 4c is a zoomed in view of Figure 4b , showing just the 24 hour window before the due date. This makes it clear that average scores significantly decrease from a high around 90 to just below 75 when started within an hour of the due date.

Plots for submit time show similar trends and are omitted due to space constraints.

The previous analysis was done using our complete data set, but we also wanted to explore whether these trends hold for each assignment type. We reproduced the plots in Figure 4 for all 14 assignment types used by our platform. We found that these trends hold for homework, quiz and exam assignments, but not all assignment types. One example where it does not hold is for LearnSmart assignments. This is most likely because Learnsmart assignments are used to drill students on a set of topics, and take a shorter period

of time to complete. Students can start them the day they are due and have plenty of time to complete satisfactorily.

4. DETERMINING DATA-DRIVEN NUDGES

We used this exploratory analysis to determine the nudge messages used in our visualizations. Based on the analysis above, conclusions could not be drawn about the day of the semester or hour of the day an assignment is submitted without further data collection and research, so these messages come from the trends seen in our exploration of start and submission time. It is clear that average scores decrease significantly as assignments are started and submitted closer to the due date and after the due date. Messages to students about when to start and when to submit are both similar in spirit, so we decided to focus our messages on starting early and avoiding submission after the due date.

We include four types of messages in our visualizations. When a student submits an assignment after the due date, they see the following message:

'Turning this assignment in late cost you <x> points! Stick to deadlines to help bump up your scores.'

and the amount of time the assignment is late is displayed in the right hand panel. When there are multiple late submissions over the semester, they will also be shown how many have been submitted late and the following additional message:

'Looks like a pattern is emerging. Better time management can help you meet deadlines.'

We also have a pair of messages focusing on starting assignments early. When students start a homework, quiz or exam within one day of the due date and they do not receive a score of 90 or better, they will receive the following message:

'Starting more than one day before the due date could result in better grades. Give yourself more time!'

If they repeatedly start assignments late, then they will see how many assignments have been started late and the following additional message:

'Late starts can lead to lower scores. Start assignments early and give yourself more time to perform better.'

These messages are designed to nudge students toward actions that will improve their performance. By providing explicit feedback about how many points they lost by submitting late, when they started assignments relative to the due date, and highlighting repeating behaviors, these message encourage students to evaluate their current actions and provide suggestions for adjusting their behavior to increase future performance on assignments.

5. CONCLUSIONS AND FUTURE WORK

In this paper we present an exploratory analysis of assignment submission data to find trends in student behavior that lead to increased performance. The data confirmed that student performance significantly decreases when assignments are started on the same day they are due and when they are submitted after the due date. We use these trends to develop data-driven nudges for students, which encourage behaviors that will help them achieve higher scores on assignments. Students see these messages when they start assignments on the same day as the due date, submit after the due date or repeatedly start or submit assignments late. These nudges are incorporated into a set of visualizations as part of our higher education platform, aimed at providing continuous, personalized feedback to students on their assignments and encouraging self-regulated learning through highlighting actionable strategies for increasing performance.

Our analysis revealed several promising avenues for future research. First, it would be interesting to understand why there are two particular assignment types that are submitted much less often than other types. This information could be used to encourage students to complete these specific assignment types or to alert instructors that these assignments are not being completed at an alarming rate and perhaps help them adjust their course to encourage completion.

We also saw potential trends in the analysis of the day of the semester assignments are completed, but we need to collect data on course start and end dates in order to clean the data set. This could lead to nudge messages reminding students to submit work as the semester progresses and scores tend to decrease. Similarly, we need to collect time zone information for each student so dates can be adjusted to local time for the analysis of the hour of the day assignments are submitted. This could lead to messages that remind students that submitting work in the early morning hours tends to lead to decreased performance.

In addition to these areas of future work, it would be interesting to do a long-term study looking at the affects of using our platform with nudge messages to understand how it affects student behaviors compared to a system that does not provide nudge messages.

6. ACKNOWLEDGMENTS

This paper is based on work supported by the McGraw-Hill Education Digital Platform Group (MHE DPG). Despite provided support, any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

7. REFERENCES

[1] D. L. Butler and P. H. Winne. Feedback and self-regulated learning: A theoretical synthesis. *Review of educational research*, 65(3):245–281, 1995.

[2] C. Carmean and P. Mizzi. The case for nudge analytics. *Educause Quarterly*, 33(4):n4, 2010.

[3] J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[4] D. J. Nicol and D. Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218, 2006.

[5] B. Wildavsky. Nudge nation: A new way to prod students into and through college. *ES Select*, 2013.

[6] B. J. Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1):3–17, 1990.

Educational Reports That Scale Across Users and Data

Rob Rolleston
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-0712
Rob.Rolleston@xerox.com

Richard Howe
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-5283
Richard.Howe@xerox.com

Mary Ann Sprague
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-2913
MaryAnn.Sprague@xerox.com

ABSTRACT
The field of education is undergoing fundamental change with the growing use of data. Fine-scale data collection at the item-response level is now possible. Xerox has developed a system that bridges the paper-to-digital divide by providing the well-established and easy-to-use paper interface to students, but digitizes the responses for scoring, validating, reporting, and managing data using a range of digital technologies. The Ignite™ system supports written responses, shading, connecting lines, multiple choice selections, and other question types. For some users, monitoring is at a very fine-grain level in both time and skill, while for others the data is used for more summative evaluations and strategic planning; one users' details may be another user's overview. All the reports presented in this document use the same basic atomic data elements and associated meta-data. The hierarchical nature of the organization of users requires that these atomic elements be combined in different ways for specialized visual representations, dependent upon the needs of the user.

Keywords

Visualization in Education, Application, Data Transformation and Representations, Field Studies, Ethnography.

1. INTRODUCTION
Technology and regulations regarding education have increased the availability of student data as well as the need to track student performance over time. No Child Left Behind [7], Race-to-the-Top [9], and the Common Core State Standards [3] are all efforts within the United States that have endeavored to make the tracking of student learning growth more measurable. Despite the increase in the access to, and need for data and data analysis, the abilities for many educators to make use of available data has not kept pace with the need [5]. Data analysis requires knowledge and tools to which not all educators have easy access [13]. Enabling data to be visualized in ways that are familiar to educators will help encourage the use of student data to inform student learning and instruction. This paper provides one example of how reporting student data in a user friendly form, for many levels of users, can help educators to find more effective uses for their student data.

2. BACKGROUND

2.1 Teachers Changing Their Instruction and Their Needs
Teachers have begun transitioning from curriculum-based instruction to student-centered instruction, which shifts the focus to assessing students at the beginning, middle and end of an instructional unit. In this way, teachers learn what the students already know about a particular subject from the start, where to focus needed instruction, and collect data throughout the process of their learning growth.

To help support the shift to utilizing student data to inform day-to-day instruction in ways that fit more closely into educator's current work processes, Xerox has created the Xerox Ignite™ Educator Support System [16]. Ignite™ is a web-based teacher tool for printing, scanning and scoring a variety of hand-marked student work and also manages the student data and produces personalized reports. Student work is generally an assessment (e.g. a quiz or test).

The item-response level of information is defined as an atomic unit: "A student is presented an item on a date by a teacher and provides some type of response." Each part of the atomic element contains additional metadata. All the reports present views of the same underlying data, but with differing levels of aggregation, dependent upon the needs of the user. This paper describes these differing user requirements and how a set of consistent and connected graphical reports can scale across the needs of these different users and their needs for data.

3. RELATED WORK
Data and data mining usage in the education domain (educational data mining or EDM) is relatively new. The field has grown rapidly for just over a decade [1].

There is a desire that the use of data will foster improvements at all levels of education. The desire for data-driven improvement in learning is countered by a concern that the use of data by itself will lead to too much of a focus on testing rather than teaching [5]. Over the years the focus of research has moved more into the field of prediction [2], and it may be that the real value will come over time when enough longitudinal data is available.

Public educational institutions have a hierarchical nature. In the United States primary schools there is a hierarchy of superintendents, principals, team leaders, and classroom teachers all making decisions. This hierarchy of users share common tasks including the analysis and visualization of data, providing feedback to support instructors, recommendations for students, and grouping of students, among others. The hierarchical nature of the users within the educational organization presents interesting challenges in both EDM [2] and in the use of the data. Teachers want easy-to-use systems, with a "desire to see assessment results at the level of subscales (groups of test items) related to specific standards and at the level of individual items in order to tailor instructions." [6] "Decisions are made at all levels of school organization. The superintendent makes decisions concerning a school district's goals and strategies. Then principals make tactical decisions concerning those goals and strategies to accomplish them in relation to their own buildings. Department heads and team leaders then make curricular and operational decisions to carry out the day-to-day activities of a department or unit. And, finally, classroom teachers make decisions in their classrooms".

Others have investigated the use of visualization in higher education situations with limited success [8]. The use of on-line learning tools has led to visualizations of curriculum [4], the design of models of student learning [10], the use of graph structures to understand

patterns of student enrolment [14] and the use of a small set of static and interactive visualizations of user data [11].

4. USERS AND USER REQUIREMENTS

4.1 Ethnographic Study of Pilot Deployment

During a technical pilot of the Xerox Ignite™ application, four elementary schools in two school districts participated. An ethnographic study was conducted to observe teachers' processes to identify challenges with using the pilot tool, and to collect user requirements and needs for the system. Observations and open-ended interviews [12] were the primary data collection methods to study teachers' assessment practices. Many teachers expressed a desire to know how their students were performing in the skills being taught, and wanted to understand how well the students understood the skills.

Talks with principals and school district administrators, including district data specialists, uncovered another level of requirements [29] relating to trend analysis, student growth over time, class-to-class and school-to-school comparisons, and progress monitoring. Principals and administrators expressed a need to see student data at the grade level, reaching from single classrooms, to all classrooms for a single grade in a single building, to entire buildings or the entire district.

Just as the assessments are used for different purposes, the users of reports have different needs. The users have been segmented into three major groups: A single teacher & class, principal or lead teacher with several classes within a school, and a district administrator looking across multiple schools in the district.

4.2 Single Teacher / Class

A teacher working with a class, or a single student in a class, is the lowest level of granularity within the current scenario. In this situation a teacher has one of two major goals: assessing the success or direction of a lesson or helping a single student.

To assess the success of a lesson plan, a teacher needs to see the class average, but also details about the mastery of different skills within that teaching unit. In Ignite™, the teacher can group and sort the questions on an assessment report according to the metadata related to the skills connected to each question.

To help an individual student, the teacher, student, and often parent need fine-scale information about specific mastery of skills. The teacher must be able to communicate with the student and parent on specific problem areas that need immediate work.

The data markers for this type of user must reveal information about the individual student, and the specific question or skill. Reports must reveal the data at the level of the basic atomic data unit, to the level of every item and response by a student to that item.

4.3 Several Classes within a Grade or School

A school principal or lead teacher within a grade or subject area needs a middle level of data aggregation. A principal is in charge of an entire school building, typically covering several grades. A lead teacher is typically focused on a single grade, or a single subject within a grade. Users at this level are typically looking at the overall progress of a cohort and the management of class or group affiliations of students. The goal is to best assign students to classes or groups and to insure that these classes or groups are on track to meet marking period and yearly goals.

The data markers for this type of user need to reveal information about the class statistics and summary information about individual students. Reports for this level can also reveal the data so that cohorts can be compared, and individual outliers within classes or groups can be identified. Skill proficiency can be shown across time and across multiple classes within a given group giving a wider view of proficiency trends. The skills are grouped at the larger unit or quarterly time intervals, and not at the individual skill code level.

4.4 Across Schools within a District

District administrators analyse data to determine long-term trends and comparisons, to report to state agencies, and to evaluate curriculum. Users at this level are more focused on summative and high-stakes assessments. These users look across schools, and compare their own district with other districts in the area and those with similar social and economic demographics.

The data markers for this class of user need to reveal information about the overall aggregate status of a district or school, with visual markers at the grade, teacher, or building level. Information about individual students is not identified or required. Trends for skills are most often limited to subject and grade level expectations.

5. DESIGN CONSIDERATIONS

Design choices were made in response to the user requirements discussed above. These design choices fell into two major areas: The general usability or workflow, and the visual attributes of the reports.

5.1 Report Selection Workflow

To produce a report, the user needs to select both a data set and a report type. These selections are not independent; as not all data can be rendered as any report, and each report needs an appropriate set of data. A linear sequential method was developed to guide the user through the report selection process.

The logical concept of the selection process is shown in Figure 1. The first step is the top row of the selection space where the user specifies "Who?" i.e. the target user and student aggregation level of the report. The second step is the left most column of the selection space where the user specifies "What?" i.e. how many assessments are to be viewed in the report. The third step of the selection process defines "When?" i.e. is this report for a single instance, or does it cover multiple recurrences. These three linear sequential steps allow the user to navigate simply through a three-dimensional specification space ending with a choice of just a few different eligible reports.

(2)What?	(1)Who? (3)When?	Teacher / Class	Principal or Lead Teacher / Several Classes	District/ Multiple Schools
Single Assessment	Once	• Matrix • Table • Image	• Matrix • Bar	• Distribution
	Over Time	• Matrix • Line	• Grouped Bar • Lines	• Lines
Portfolio of Assessments	Once	• Bar	• Bar/ Scatter	• Distribution
	Over Time	• Lines	• Lines	

Figure 1 - Report selection table

The layout of the report selection table shown in Figure 1 also aids in understanding the automatic aggregation of data according to user and data selection. The different user views for the case of aggregating data about a single assessment given once are shown in Figure 2. In all cases, that same single assessment is chosen, but the report is different depending upon the scope of classes selected. A teacher meeting with a single student is most interested in the report

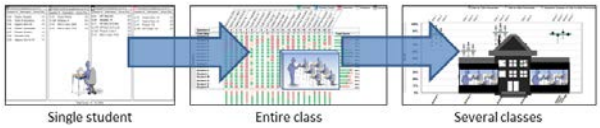


Figure 2 Flow of data aggregation and report type for a single assessment given once. In moving from left to right the question detail is lost and the higher level averages are presented.

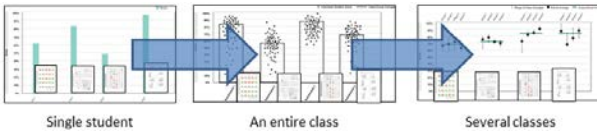


Figure 3 Flow of data aggregation and report type for a portfolio of four assessments. In moving from left to right the student specific information is lost and higher level distributions are presented.

on the left; a table report that provides detailed information about each question. A teacher assessing the progress of an entire class is most interested in the report in the middle; a matrix heat map. A district administrator, looking at several classes is most interested in the report on the right; a distribution report that provides information about school and class averages, and some notion of the distribution of scores within each class and range of scores within each school.

The different user views for the case of aggregating data about a portfolio of assessments are shown in Figure 3. In all cases, the same set of assessments is chosen, but the reports are different depending upon the scope of classes selected. A teacher meeting with a student and parent is most interested in the report on the left; a single bar chart report that provides a student's performance across a range of skills, at a single point in time. A group of teachers meeting to discuss the grouping of students is most interested in the report in the middle; a bar chart report that provides information about the class average, but also the score of each student. A district administrator, looking at several classes is most interested in the report on the right; a distribution report that provides information about school and class averages, and some notion of the range of scores within a class.

6. REPORT DESIGNS

There are 6 different basic report styles: (1) image, (2) table, (3) matrix heat map, (4) line, (5) bar, and (6) distribution. Where appropriate, report styles were customizable to properly display the defined data set selected by the user.

6.1 Image

The image report is an image of the scanned and validated assessment. This report is the only portrait mode report, and has no header or footer information to maximize the actual assessment image region. Two examples of this report type are in Figure 4.

The visual representation exactly matches the physical paper version of the assessment.

The green highlight areas are those questions that were validated by the teacher as correct. Responses that were validated as incorrect were shown with a red overlay, skipped responses with a yellow

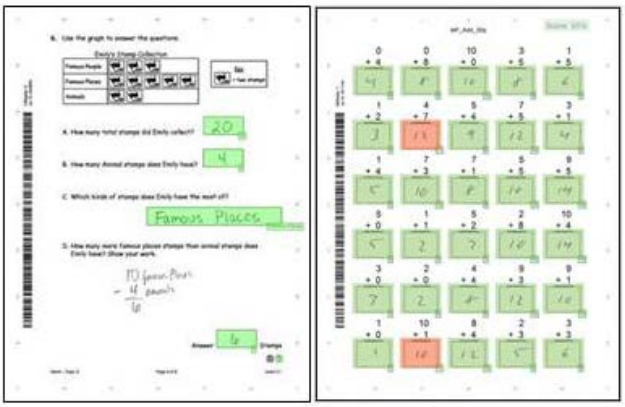


Figure 4 - Image reports where green indicates the area of a correct answer and red is an incorrect answer

overlay, and those validated for partial credit were shown in blue. The color coding served as a pre-attentive signal to the correctness of the student's answer.

This report is used by a teacher in individual consultations with a student and/or parent. It is a record of the marks the student made on the assessment, and how the teacher validated each question.

6.2 Table

The table report is a listing of the correct, partial, skipped, and incorrect responses to a single assessment, by a single student. The questions are placed in the column corresponding to the scored and validated response and include the question number, the question description (obtained from metadata), the number of points the student earned for the question, and the total number of points possible on the question. The data is presented at the fine-scale atomic level, with the use of metadata. By using the metadata about the question, the report is applicable to any type of question. If the question description metadata is used to encode specific skill information, then a quick visual scan down the list reveals common skills that have appeared in a single column, or the columns could be sorted according to some value(s) in the metadata.

This report is used by teachers with students and parents. It is easy for students to understand that their goal is to make all the questions appear in the left-most (correct) column.

6.3 Matrix heat map

The matrix heat map report is a visual summary of the responses to each question on an assessment, by each student. The data for each student appears in a row, and the data for each question appears in a column. At each intersection point there is a graphical representation of the student's response to that question.

Sorting the rows and columns of the matrix provides the user with a quick visual assessment of several different stories [15]. By sorting the rows in order of student score, the teacher is able to quickly discern who has mastered the skills, and who has not as well as identifying which areas where most students are struggling. By sorting the columns in order of the correct number of student responses, or sorting by the metadata associated with questions, the teacher can quickly see what groups of questions or skill sets were successfully mastered by the class and which were not. If a particular set of question were not mastered by the class, the teacher now has this additional piece of information and can decide upon the need to re-teach a particular set of skills to that subgroup of students.

6.4 Bar

The Bar Chart is used in several instances. For consistency, all bar charts use the vertical axis to represent score. The horizontal axis can be categorical values of assessments, students, classes, or dates. The

bars can be singular or grouped, depending upon the amount of data to be displayed.

6.4.1 Comparing Students or Classes over time

The collection of scores for a single assessment repeated over time can be represented using a bar chart. In this case, the x-axis is a categorical list of students or classes. There is a group of bars for each student or class, and a bar of each instance in time.

6.4.2 Portfolio of Assessments

The portfolio of assessment scores for a single student, a single class, or all the classes within a school all use similar representation. The common representation provides users with a common mental model for scaling across such aggregations. In all these cases, the x-axis is a categorical list of assessments.

For the case of a single student, there is one bar for each assessment.

For the case of a single class, there is one bar for each class average score, and the distribution of student scores are overlaid as a jittered scatter plot, where the x-jitter is bounded to the width of the corresponding bar

6.5 Line

The line chart is used only for trends over time. For consistency, all line charts use the vertical axis to represent score. For a single assessment, the score can be absolute or percentage, and may be aggregated at the student, class, or school level. In all cases, the x-axis is a categorical list of times, e.g. the date each assessment was given. These reports are used to view the progress over time of one or more assessments.

When viewing the portfolio of results over time, each line represents a different assessment. The points on the line are the relative score achieved on the given assessment at that point in time. The score can be a single student, aggregated over a class, or over a school.

6.6 Distribution

Reports showing a statistical summary of distributions typically are used only by district level users. These types of reports use the greatest amount of aggregation of atomic data elements. For consistency, all distribution charts use the vertical axis to represent score. The x-axis is a categorical list of assessments or schools. The markers are groupings of box and whisker plots.

7. CONCLUSION

We designed and deployed a system that implements a large and comprehensive set of reports for use by educators at different levels. The system was designed based on ethnographic studies and iterative participatory feedback from users, as well as subject matter experts on staff. The novel aspect of this work was the creation of a complete system that bridges the paper-digital divide, offers views into the data at different levels of granularity and aggregation, and scales to match a user's needs and work processes while preserving similarity in the selection workflow and report design.

The reports all used the same underlying fine-grain data element at the item-response level by each student, but aggregate the data differently dependent on the user's needs. Users included teachers-students-parents, lead teachers or principals, and district level administrators. These users had needs that required scalability thru several levels of data aggregation.

Report designs focused on the re-use of basic design concepts across the different visual representations, thus allowing users to more easily traverse the report space by learning a common set of patterns and styles. The report selection process follows a linear progression of selecting the collection of students, the collection of item-responses, the time, and finally any visual representation options for the data.

8. ACKNOWLEDGMENTS

The authors wish to thank the Ignite™ school support team of Todd Conrow, Deb Drago-Leaf, and Lori Schirmer for their critiques, suggestions and detailed knowledge of what it is really like to be in the classroom.

9. REFERENCES

[1] A. Peña-Ayala. (2013). "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Systems with Applications. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.08.042.

[2] R.S.J.D. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," *Journal of Educational Data Mining*, Article 1, Vol. 1, No. 1, Fall 2009.

[3] "Common Core State Standards Initiative." [Online]. Available: http://www.corestandards.org/2011.

[4] M. Ida, "Web Service and Visualization for Higher Education Information Providing Service," 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS), 16-18 July 2010, pp. 415 – 418. Available: 10.1109/ICSESS.2010.5552349.

[5] M.M. Kennedy, "Data Use by Teachers: Productive Improvement or Panacea?" *Education Policy Center at Michigan State University, Working Paper #19*, May 20, 2011.

[6] B. Means, C. Padilla, and L. Gallagher, "Use of Education Data at the Local Level, From Accountability to Instructional Improvement," *U.S. Dept. of Education*, January 2010. Available: https://www2.ed.gov/rschstat/eval/tech/use-of-education-data/use-of-education-data.pdf.

[7] "No Child Left Behind" [Online]. Available: http://www2.ed.gov/nclb/landing.jhtml.

[8] M. Pinto, R. Raposo, and F. Ramos, "Comparison of Emerging Information Visualization Tools for Higher Education," *2012 16th International Conference on Information Visualisation*. Available: 10.1109/IV.2012.27.

[9] Race to the Top, http://www2.ed.gov/programs/racetothetop/index.html

[10] M. Sasakura and S. Yamasaki, "A Framework for Adaptive e-Learning Systems in Higher Education with Information Visualization," 11th International Conference Information Visualization, 2007.

[11] K. Silius, A.-M. Tervakari, and M. Kailanto, "Visualizations of User Data in a Social Media Enhanced Web-based Environment in Higher Education, "*2013 IEEE Global Engineering Education Conference (EDUCON)*, pp. 893 – 899.

[12] M.A. Sprague and M. Fuhrmann, "Ethnography Supports Changes to Student-Centered Instruction," *TQR 4th Annual Conference*, January 18, 2013.

[13] J.C. Wayman, "Involving Teachers in Data-Driven Decision Making: Using Computer Data Systems to Support Teacher Inquiry and Reflection," *JOURNAL OF EDUCATION FOR STUDENTS PLACED AT RISK*, 10(3), 2005, pp. 295–308.

[14] D. Wortman and P. Rheingans, "Visualizing Trends in Student Performance Across Computer Science Courses, "Proceedings *of the 38th SIGCSE technical symposium on Computer science education*, 2007, pp. 430 – 434.

[15] Journal of Educational Data Mining, http://www.educationaldatamining.org/JEDM/index.php/JEDM/issue/archive

[16] Ignite, http://www.acs-inc.com/health-and-human-services/education-services/ignite-teacher-tool.aspx

Mining Login Data For Actionable Student Insight

Lalitha Agnihotri
McGraw-Hill Education
lalitha.agnihotri
@mheducation.com

Ani Aghababyan
McGraw-Hill Education
ani.aghababyan
@mheducation.com

Shirin Mojarad
McGraw-Hill Education
shirin.mojarad
@mheducation.com

Mark Riedesel
McGraw-Hill Education

Alfred Essa
McGraw-Hill Education

ABSTRACT

Student login data is a key resource for gaining insight into their learning experience. However, the scale and the complexity of this data necessitate a thorough exploration to identify potential actionable insights, thus rendering it less valuable compared to student achievement data. To compensate for the underestimation of login data importance, in this paper we performed an exploratory data analysis of a large educational dataset consisting of 100 million instances of login data from 1.5 million unique students who attempted 783 thousand assignments. The data were from a McGraw-Hill Education web-based assessment platforms called *Connect*. Different data mining methods were employed to answers our initial questions regarding students' login behavior. Most of the findings were consistent with the intuitive expectations of student login patterns such as a considerable decline of activity on Saturdays, a visible peak on Sunday evenings, a high activity in September and February, and an increased activity toward later hours of the day. However, we also discovered an unexpected result while investigating the effects of the login activity, the performance scores, and the attempts. Surprisingly, this analysis showed a high positive correlation between login activity and performance scores, only up to a certain threshold. This provided us a new hypothesis on student groupings, which we explored through a cluster analysis. As a result of our exploratory efforts, a significant amount of patterns emerged that not only confirmed previously set forth expectations but also provided us new hypotheses, which can be leveraged to improve student outcomes.

Keywords

Exploratory data mining, assessment platform, clustering, log data, pattern & trend mining

1. INTRODUCTION

An increasing number of higher education institutions are incorporating online course management platforms, which creates a tremendous opportunity for monitoring learners' academic activity. These web-based learning environments capture immense amounts of login data that could be used for student monitoring and profiling ([11]). Educational literature suggests that monitoring students' academic activity is a key to a more effective and higher quality education ([2], [3], [7], [8]). Furthermore, research shows that college students would benefit from opportunities of introspection and cognitive monitoring of their progress in order to engage in careful academic planning ([1]). Hence, given its scale, these login data are a promising resource for shedding light onto students' academic behavior.

In this paper we explore login data from a McGraw-Hill Education's (henceforth MHE) web-based assessment platform. These data can serve as a basis for instructors' personalized intervention programs and feedback for student efforts toward self-regulated learning. While interest in login data analysis has been continuously increasing, there is no standardized way of analyzing this type of data ([9]) due to diversity of the data and uniqueness of research questions. Hence, we conducted exploratory data analysis without setting a priori limitations or hypotheses on our data. In Sections 2 through 4, we discuss our methods with detailed descriptions and their findings. Section 5 contains discussions about our results and conclusions along with future work.

2. METHODS

2.1 Participants and Materials

Our research data is collected via one of the MHE assessment platforms called Connect (`http://connect.mheducation.com`). Connect is a higher education web-based assessment and assignment platform, which provides students an online environment to do their coursework and logs user activity in order to provide feedback and support to its user needs.

In this paper we explored 100 million instances of user login data obtained from Connect between June of 2013 and June of 2014. For this analysis, we used data such as students' login dates, total number of logins, number of attempts on an assignment and assignment score. Depending on the analysis, some of these data were aggregated based on time or grouped by the unique students.

2.2 Procedures and Methods

To extract the necessary data for our analyses, we used Oracle's procedural language extension for SQL (i.e., PL/SQL) [4] and Python programming language [13], along with the necessary Python libraries to query, wrangle, clean, plot, and explore our login data. Our data contains the following attributes: student related data (e.g., student ID, student logins) and assessment related data (e.g., number of attempts, assessment score, number of attempts).

3. LOGIN BEHAVIOR ANALYTICS

3.1 Login Behavior

In this section we investigated the trends related to student logins. Figure 1 visualizes the overall pattern of student logins over the days of the week. The red line shows the average number of logins for any given day. This analysis validates the expected pattern of decreasing activity on Saturdays and increasing activity on Sunday evenings. This shows students' tendency to stay away from their homework assignments on the weekend until late Sunday when they attempt to prepare for the week. This finding is not surprising, in fact, it confirms the intuitive expectation of student academic activity on weekends vs. weekdays. If investigated further (i.e., A/B testing), this information could provide a basis for notifying students with customized and timely recommendations via Connect.

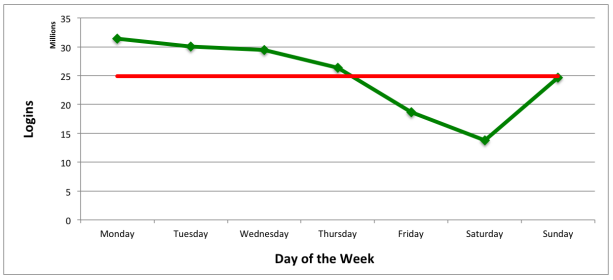


Figure 1: Logins by the day of the week. X-axis = Day of the week from Monday to Sunday; y-axis = Logins (in millions).

Next, in Figure 2 we investigated the number of logins per day. While the overall pattern of logins increasing in Fall through Spring and decreasing in Summer seemed very reasonable, the significant spike in Spring of 2014 seemed out of ordinary. To understand this unusual pattern, we requested more information from the Connect marketing team who explained that the spike in the Spring of 2014 is congruent with the new marketing effort making Connect assignments mandatory portion of students' coursework. This finding provided a data grounded confirmation of Connect team's marketing efforts.

4. PATTERN MINING & STUDENT PROFILING

For the analyses in this section, we used the average number of logins per assignment (henceforth logins), the average score per student (henceforth score), and the average attempt per assignment (henceforth attempt). In this section, we present our analysis of comparing the student login data with students' scores on assignments.

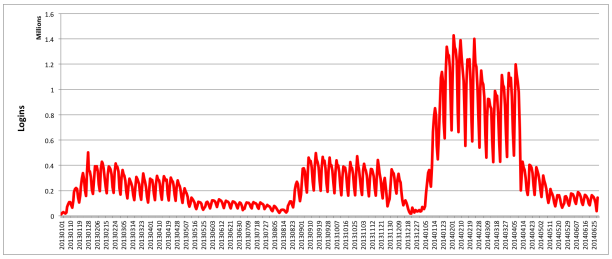


Figure 2: Logins by the month. X-axis = Days in months from 01/01/2013 to 06/25/2014; y-axis = Logins (in millions).

4.1 Login vs. Score Trends

To continue our data explorations, we decided to further investigate the potential patterns in the student login and student assignment score data.

4.1.1 Data Preparation

For this analysis, we looked at a total of 1.5 million users' assignments scored between June 2013 and June 2014. For each user, score, login and total number of attempts were normalized against users' total number of activities. Further, we eliminated some of the outliers by excluding the users with 1 or no attempts and eliminated users with more than average 50 logins which removed 100,000 users' data. On average, students login 5.5 times, have 1.03 attempts and have a score of 53% per activity.

4.1.2 Data Analysis

We plotted student logins per assignment vs. student's median score (see the green line in Figure 3). In this plot, we used the median score instead of the mean of the scores in order to account for the high variability of the distribution of scores. This figure shows that student median score grows as the number of logins increases. However, after a certain threshold, the score tends to decrease as the number of logins per assignment increases, thus showing the counterproductivity of the login activity. This contradicts to the intuitive assumption that more logins result in a better academic performance.

To further explore the relationship between login and scores, we performed a piecewise linear regression to identify possible segments in the data. Fitting a single regression line, the standard error (SE) of estimate with one regression line was σ_{est}=18. The SE for a model with two regression lines resulted in σ_{est}=12.5. We also tried fitting three regression lines through, which resulted in a higher SE of σ_{est}=16.8. Therefore, we used a model with two regression lines (see Figure 3). This resulted in a break at i=4 (i.e., Segment 1 = 0:4 and Segment 2 = 5:50). This suggests two distinct segments in the data. In the first segment, as the number of logins increase, the performance improves (slope = 6.48; correlation = 0.99). However, after a certain threshold, 4 logins, the scores plateaus, and gradually decrease as the logins increase (slope = -0.45; correlation = -0.93). This hypothesis is further explored in the next section through cluster analysis.

4.2 Student profiling

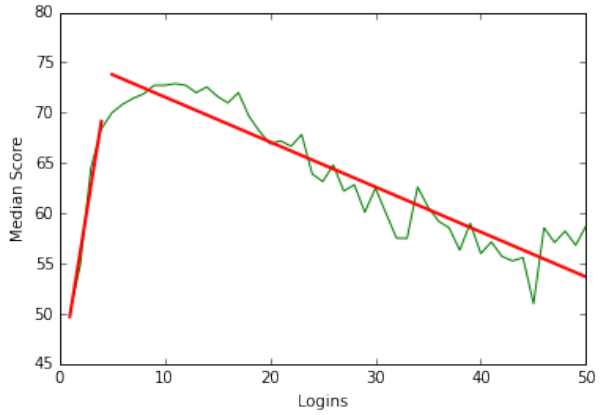


Figure 3: Piecewise linear model. X-axis = Number of logins per assignment; y-axis = Median score.

4.2.1 *K-Means Clustering Method*

Following the hypthesis formed in the previous section, we explored student login patterns through k-means clustering. In k-means clustering, data is partitioned into k clusters where each observation is assigned to the cluster with the nearest mean ([6]). The clustering process starts by choosing k random observations as initial cluster centroids. Thereafter, each observation is assigned to the nearest centroid and the new centroids are recalculated using the average of the data points in each cluster. We selected Euclidean distance as the distance metric in k-means clustering ([5]) where within-cluster sum of squares (hereafter, WCSS) is the cost function. Representing the data as a set of N observations $\{x_1, x_2, , x_n\}$, where each observation is a D-dimensional vector of D attributes, k-means clustering partitions N observations into k clusters $\{c_1, c_2, , c_k\}$ where WCSS is minimized as:

$$argmin \sum_{k=1}^{K} \sum_{X \in c_k} \parallel X - \mu_k \parallel^2$$

where μk is the mean of points in c_k. To accommodate the scale of our dataset, we have selected k-means clustering method due to its computational speed and efficiency compared to hierarchical clustering. In addition, k-means clustering is a robust approach, which results in non-overlapping clusters that are very easy to interpret. We have used the Elbow method ([12]) to identify the optimal number of clusters. In this method, average WCSS is measured as the number of clusters increase. Having more clusters results in smaller distances from centroids and hence a smaller average WCSS. However, the amount of drop is not constant as the number of clusters increase and the decrease in average WCSS flattens at a certain k value. This value, called the elbow metric, creates a break in the elbow graph and is a good measure for identifying optimal number of clusters.

4.2.2 *Clustering Results*

In this analysis, we used the same data aggregations for students' login, score and attempts as described in the beginning of this section to explore student groupings according to their login behavior. The elbow method is used to decide an optimum number of clusters. Figure 4 shows the average WCSS value as the number of clusters increases from 1 to 9. The graph nearly flattens after k equals to three, thus suggesting 3 as the optimal number of clusters.

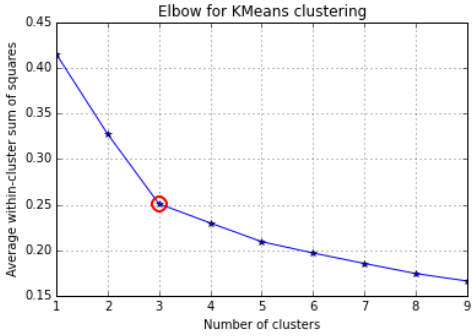


Figure 4: Elbow metric. k=3; x-axis = Number of clusters; y-axis = Average WCSS.

We used Scikit-learn python library ([10]) to implement k-means clustering. Figure 5 shows a 3D scatter plot of the three attributes used to cluster the data where the data points are colored by the cluster labels. Figure 5 shows

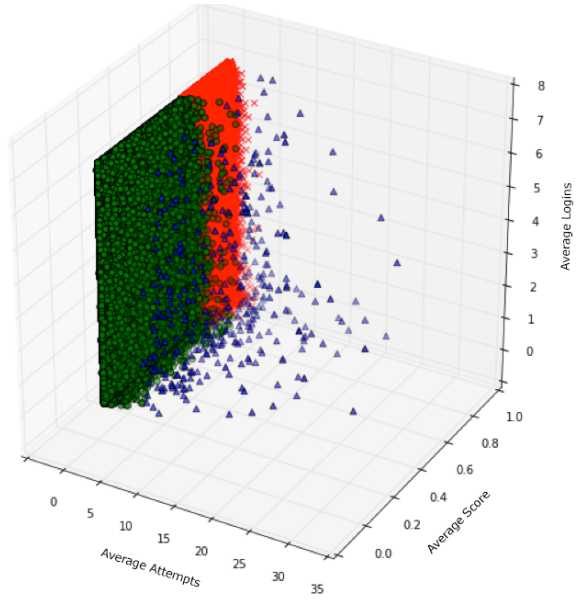


Figure 5: 3D scatter plot. Cluster 1 (red) = High Achievers; Cluster 2 (green) = Low Achievers; Cluster 3 (blue) = Persistent Students; Attempts = x axis; Logins = y axis; Score = z axis.

three sets of distinct student login profiles. The Cluster 1 (red), whom we label as *High Achievers*, represent a group of students with a low number of attempts, a medium number of logins, and a high score. The Cluster 2 (green), whom we label as *Low Achievers*, is the group with a medium number of attempts, and low number of both logins and score. Finally, the Cluster 3 (blue), whom we label as *Persistent Students*, is the most distinct group with a high number of both attempts and logins, and a medium score. To quantify

this information, in Table 1 we have tabulated the count, the mean and the standard deviation of these three attributes across each of the three clusters. In addition, we have simplified this content in Table 2.

Table 1: Cluster Statistics. Total = number of observations. SD = standard deviation.

		Average Attempts	Average Score	Average Login
Cluster 1 (High Achievers)	Count	1097675	1097675	1097675
	Mean	1.03	0.84	8.51
	SD	0.19	0.37	58.07
Cluster 2 (Low Achievers)	Count	780405	780405	780405
	Mean	1.05	0.24	6.03
	SD	0.25	0.17	21.33
Cluster 3 (Persistent Students)	Count	1220	1220	1220
	Mean	9.82	0.56	35.65
	SD	6.83	0.32	62.13

Table 2: Student groups based on cluster statistics.

	Login	Attempt	Score
High Achiever	Medium	Low	High
Low Achiever	Low	Medium	Low
Persistent Student	High	High	Medium

Table 2 shows that Cluster 1 (high achievers) includes students with the highest score among the three clusters. Low achievers, Cluster 2, stand out with a very low score and a low number of logins. This shows a relationship between the low logins and the low performance scores in students with very high or very low scores. However, students with medium score have very high average logins and high average attempts per activity. This fluctuation between average score and login indicates a non-linear and non-trivial relationship between student behavior (number of logins and attempts) and performance.

5. CONCLUSION & DISCUSSION

In this paper we explored student login data collected from MHE's Connect higher education platform. The investigation of student login activity reveals a non-linear relationship between student activity and performance. Piecewise linear regression revealed that students who do better on their assignments tend to login more. However, if a student logs in 5 or more times per assignment, their performance tends to plateau and then deteriorate. Thus, it would be beneficial for the instructor to intervene at this point as it might indicate that the student has not grasped the concepts required for the assignment. Finally, investigating student login behavior led to identifying three distinct groups of students: high achievers who login just optimum number of times to get high score, low achievers, who login very rarely and tend not to do well, and persistent students who show grit in their efforts to succeed by logging in and attempting the most but still perform less than high achievers. The educational value of such finding is in identifying and encouraging certain activity behaviors that are correlated with good performance.

Future work will be concentrating on factors such as the variability in the students' scores based on the due date of the assignments, time spent on assignments, potential recommendations or instructors actions and effectiveness of these recommendations via A/B testing. Finally, we will be attempting to join students academic performance gathered from Connect to their performance or other institutional or demographic data in order to predict student academic success.

6. ACKNOWLEDGEMENTS

This paper is based on work supported by McGraw-Hill Education. We would like to extend our appreciation for all the informational support provided by the Connect Team, the research support provided by the MHE CDO Stephen Laster, and the Analytics team at DGP. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

References

[1] N. E. Commander and B. D. Smith. Learning logs: A tool for cognitive monitoring. *Journal of Adolescent & Adult Literacy*, 39(6):446–453, 1996.

[2] K. Cotton. Classroom questioning. *School improvement research series*, 3, 2001.

[3] R. DuFour. Professional learning communities. 1998.

[4] S. Feuerstein and B. Pribyl. *Oracle pl/sql Programming.* " O'Reilly Media, Inc.", 2005.

[5] J. C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[6] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[7] K. Leithwood, K. Seashore Louis, S. Anderson, K. Wahlstrom, et al. Review of research: How leadership influences student learning. 2004.

[8] R. Mazza and V. Dimitrova. Visualising student tracking data to support instructors in web-based distance education. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 154–161. ACM, 2004.

[9] M. Muehlenbrock. Automatic action analysis in an interactive learning environment. In *The 12th international conference on artificial intelligence in education, AIED*, pages 73–80, 2005.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[12] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[13] G. Van Rossum and F. L. Drake. *The python language reference manual.* Network Theory Ltd., 2011.

Building Models to Predict Hint-or-Attempt Actions of Students

Francisco Enrique Vicente Castro **Seth Adjei** **Tyler Colombo** **Neil Heffernan**
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609, USA
{fgcastro, saadjei, tjcolombo nth}@wpi.edu

ABSTRACT
A great deal of research in educational data mining is geared towards predicting student performance. Bayesian Knowledge Tracing, Performance Factors Analysis, and the different variations of these have been introduced and have had some success at predicting student knowledge. It is worth noting, however, that very little has been done to determine what a student's first course of action will be when dealing with a problem, which may include attempting the problem or asking for help. Even though learner "course of actions" have been studied, it has mostly been used to predict correctness in succeeding problems. In this study, we present initial attempts at building models that utilize student action information: (a) the number of attempts taken and hints requested, and (b) history backtracks of hint request behavior, both of these are used to predict a student's first course of action when working with problems in the ASSISTments tutoring system. Experimental results show that the models have reliable predictive accuracy when predicting students' first course of action on the next problem.

Author Keywords
Educational data mining; intelligent tutoring systems; student modeling; student behavior.

1. INTRODUCTION
Most educational data mining (EDM) research focus on modeling student behavior and performance. Algorithms such as Bayesian Knowledge Tracing [1], and Performance Factors Analysis [4] have been used to achieve this end. In intelligent tutoring systems, it is crucial to be able to understand student behavior to provide better tutoring practices and improved content selection for these systems. Student behavior may provide another means to identify low-knowledge or low-performing students and determine when to proactively intervene. Previous works show that students who are more likely to ask for help on problems learn less and perform less. A study on students' help-seeking behavior in an SQL tutoring system [3] suggests that students who used help very frequently had the lowest learning rate and had shallow learning. A study that used the sequence of attempts and hint requests to predict student correctness found that students who first made attempts on problems performed better than those who requested for help first [2]. The Assistance Model [6] used the number of hints and attempts a student needed to answer a previous question to predict student performance. Gaining the capability to recognize students' need for assistance ahead of time by looking at students' pattern of actions could lead to more proactive interventions, such as identifying prerequisite skills, adapting pedagogical methodologies, or gaining insight on student problem solving methodologies.

With these in mind, we then ask: how do we determine when students will ask for help when using an ITS? On the exploratory level of model development, what information may be useful for developing models that forecast students' need for assistance? In this work, we define two models that use information on problem attempts and help requests used by students in the ASSISTments tutoring system: (1) *Attempt/Hint Count model* (AHC) makes use of information on the number of attempts and hints used by students on a question to predict the occurrence of a help request as the first action on the next problem, and (2) *Hint History model* (HH) makes use of the history of hint request as the first action in preceding questions to predict the occurrence of a help request as the first action on the next problem.

We utilized tabling methods to generate prediction values from the information used by each model. Tabling methods have been found to be effective alternatives for performing predictions using datasets and offer the advantage of being computationally inexpensive and easily expandable to leverage more features into simple models [2, 7].

2. DATASET
The data used in the analysis is from ASSISTments, an online tutoring system maintained at the Worcester Polytechnic Institute that provides tutorial assistance if students make incorrect attempts or ask for help [5]. The dataset is from released ASSISTments data that spans about five months within the 2012-2013 school year, containing

599,368 student log entries. More details about ASSISTments data can be accessed from: https://sites.google.com/site/assistmentsdata/how-to-interpret.

Analysis for the AHC model was done on problem logs with 1 to 5 attempts taken in answering problems, accounting for 98% of all data entries (585,926 rows). Problem entries with 3, 4, and 5 available hints (AvH) were used and these accounted for 70% of the data (415,895 rows). The resulting dataset contains 420 problem sets and 12,966 students, totaling to 299,968 entries. The resulting dataset was separated into problem groups that differed in the number of available hints to avoid comparing the hint request behavior of students who had more opportunities to hint against students with fewer opportunities to do so.

Problem Group	Problem Sets	Students	Dataset entries
3 AvH	285	11,402	169,100
4 AvH	224	10,282	111,754
5 AvH	60	4,724	19,114

Table 1. AHC dataset for each of the problem groups

For the HH model, we selected entries in the dataset where each student sequence had at least 4 rows. The student sequence is the sequence of problems that a student answered. Sequences had to at least have 4 rows for the HH model which looks at the history of hint use, 3 problems prior the next problem. The resulting dataset contained 279,925 entries with 555 problem sets and 12,429 students.

3. STUDENT ACTION MODELS

In ASSISTments, students exhibit varying behaviors when encountering problems: submitting an answer to a problem first ("attempting the problem"), asking for help (hint) first, asking for hints after an initial attempt, alternating between attempts and requests for hints, or continuously attempting a problem until a correct answer has been submitted. These behaviors have likewise been observed in [2].

3.1 Initial Experiments: AHC

The AHC prediction table maps the number of attempts and hints used to the probability that the student attempted or asked for a hint on the next problem. The probability is the percentage of students who asked for a hint on the next problem. Table 2 shows a sample prediction table from training data. Table 3 shows a matching scenario using Table 2. A value under *Hints Taken* in Table 2 such as 2/3 indicates that a student used 2 out of 3 available hints for the problem and values on the first column indicate the count of attempts. Five-fold cross validation was used to train and test the AHC model on the three problem groups. Problem set and student-level analyses were done to see whether the model generalizes across unseen problem sets and students.

3.2 Secondary Experiment: HH

For HH analysis, the prediction table was generated by using the percentage of hint use as first action in three previous problems. Table 4 shows a prediction table from training data. Column labels correspond to the number of times the first action was an attempt on the problem or a hint request. For example, 1H/2A indicates that in three prior problems, a total of 1 hint as first action and 2 attempts as first action were used. Counts of attempts and hints as first action were then generated for each column. In the table, for those who used a total of 2 hints and 1 attempt in three previous problems, there are 3330 instances of attempts and 1833 instances of hint requests as first action on the next problem. *% Hint* is the percentage of instances of hint use within the bin. Problem set and student-level five-fold cross validation was used to train and test the HH model.

Attempts Taken	Hints Taken 0 / 3	1 / 3	2 / 3	3 / 3
1	0.0211	0.1001	0.2213	0.4025
2	0.0261	0.0558	0.0747	0.1105
3	0.0237	0.0447	0.0737	0.0916
4	0.0363	0.0287	0.0743	0.0949
5	0.0132	0.0263	0.0857	0.0912

Table 2. AHC Prediction Table

Student	A_C	H_C	H_T	FANP
92677	1	0	3	0.0211
92680	2	3	3	0.1105

Table 3. Matching scenario using Table 2 (Note: A_C = Attempt Count, H_C = Hint Count, H_T = Hint Total, FANP = First Action Next Problem)

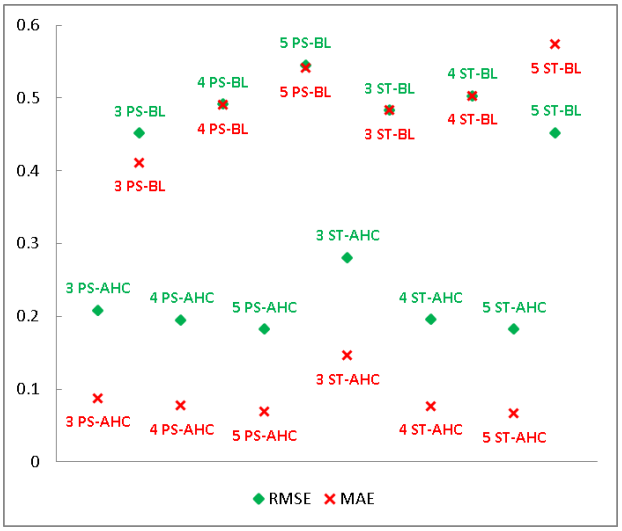
	Previous 3 First Action Hints / Attempts 0H / 3A	1H / 2A	2H / 1A	3H / 0A
# Attempt	111017	17219	3330	683
# Hint	5859	3254	1833	1663
% Hint	0.0501	0.1589	0.3550	0.7089

Table 4. HH Prediction Table

To analyze whether the number of history points affected the predictive power of HH, an additional analysis with four problems prior the next problem was done.

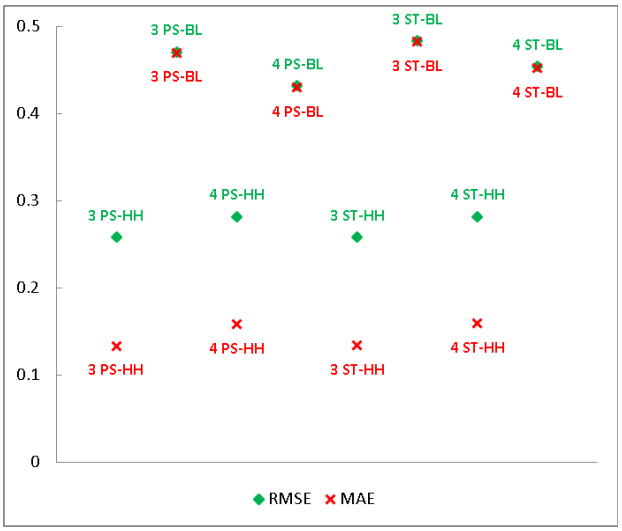
4. RESULTS AND DISCUSSION

The predictive performance of the AHC and HH models were evaluated using root mean squared error (RMSE), mean absolute error (MAE), and area under the ROC curve (AUC). Additionally, a naïve baseline (BL) model was generated for comparison, as we have found no other gold standard model for first-course-of-action prediction to compare our work with. The BL model uses the percentage of hint instances on the students' second action on all problems in the dataset. Table 5 shows a scenario for BL prediction. *Hint %* is the percentage of hint instances in the problem entries, which translates to a prediction on the students' first action on the next problem. If a student's second action on the current problem is a hint, the prediction for FANP is *Hint %*, otherwise, use *Attempt %*. The intuition for this is the hypothesis that students who have greater tendency to ask for hints on succeeding actions may most likely ask for hints in succeeding problems.



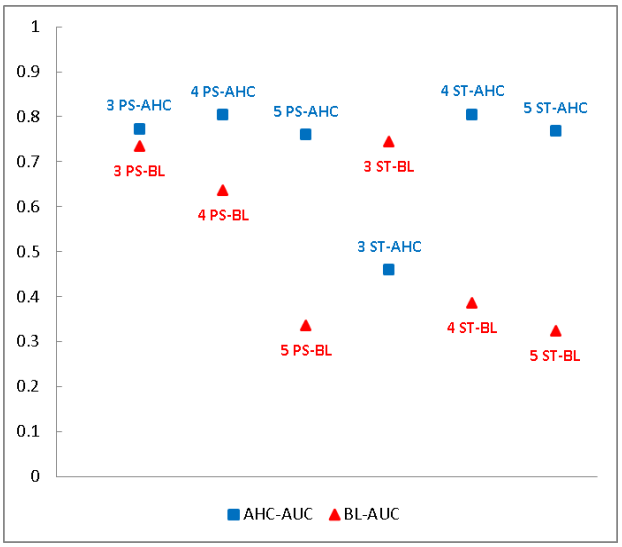
PS	3 AHC	3 BL	4 AHC	4 BL	5 AHC	5 BL
RMSE	0.2075	0.4506	0.1942	0.4910	0.1813	0.5445
MAE	0.0866	0.4104	0.0763	0.4899	0.0677	0.5403
ST	**3 AHC**	**3 BL**	**4 AHC**	**4 BL**	**5 AHC**	**5 BL**
RMSE	0.2799	0.4826	0.1945	0.5023	0.1811	0.4514
MAE	0.1452	0.4821	0.0758	0.5022	0.0653	0.5729

a. RMSE and MAE performance for AHC vs. BL across three problem groups (3, 4, and 5 available hints)



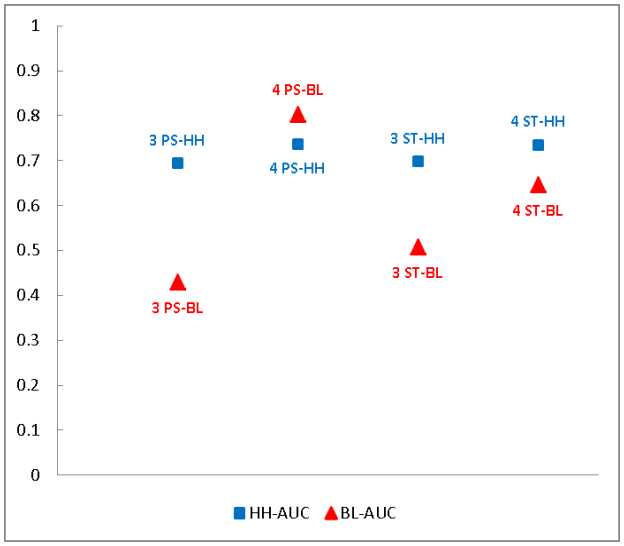
PS	3 HH	3 BL	4 HH	4 BL
RMSE	0.2574	0.4697	0.2809	0.4307
MAE	0.1327	0.4687	0.1572	0.4291
ST	**3 HH**	**3 BL**	**4 HH**	**4 BL**
RMSE	0.2573	0.4821	0.2808	0.4528
MAE	0.1328	0.4810	0.1580	0.4513

b. RMSE and MAE performance for HH vs. BL for 3 and 4 prior problems



PS	3 AHC	3 BL	4 AHC	4 BL	5 AHC	5 BL
AUC	0.7737	0.7332	0.8043	0.6338	0.7602	0.3338
ST	**3 AHC**	**3 BL**	**4 AHC**	**4 BL**	**5 AHC**	**5 BL**
AUC	0.4599	0.7419	0.8056	0.3841	0.7689	0.3223

c. AUC performance for AHC vs. BL across three problem groups (3, 4, and 5 available hints)



PS	3 HH	3 BL	4 HH	4 BL
AUC	0.6936	0.4298	0.7357	0.8026
ST	**3 HH**	**3 BL**	**4 HH**	**4 BL**
AUC	0.6989	0.5071	0.7355	0.6458

d. AUC performance for HH vs. BL for 3 and 4 prior problems

Figure 1. Problem set (PS) and student (ST) level RMSE and MAE performance for AHC, HH, and BL (a and b); Problem set and student level AUC performance for AHC, HH, and BL (c and d).

Problem entries	Hint Count: 2nd Action	Hint % (BL)	Attempt %
2200	852	0.3872	0.6127

Table 5. Sample scenario for BL prediction values

4.1 AHC Analysis

Problem set level findings for both AHC and BL are presented in Figure 1a. AHC consistently outperforms BL across all problem groups in both RMSE and MAE. Lower values for both metrics indicate better model fit. A reliability analysis to compare AHC with BL using a two-tailed paired t-test indicates that the findings are reliably different across all problem groups (p=0). The effectiveness of the model is likewise seen using the AUC metric (Figure 1c). AUC values closer to 1 indicate better model fit. It can be noted that AHC performance in all metrics are closely consistent, suggesting that the model is fairly generalizable across problems with varying numbers of hint availability. Predictive performance using student level analysis for problems with 4 and 5 available hints is fairly consistent across all three metrics; however, the model does not perform as well for problems with 3 available hints, suggesting that AHC may be used to predict the hint request behavior of unseen students, provided there is a high number of opportunities to ask for help. BL performance fails to improve as the number of available hints increase for both problem set and student-level analyses.

4.2 HH Analysis

A problem set level analysis of the HH model across the number of prior history points demonstrates that the HH model maintains a fairly consistent level of predictive performance across all three metrics. While HH significantly outperforms BL in MAE and RMSE, it is outperformed by the latter in AUC for 4 history points. This may be because the ordering of values in BL's predictions is not as close to the actual as those of HH. This situation rarely happens; we may have to try another dataset to confirm this behavior. On a student level analysis, HH outperforms BL across all values of first action prior history points (Figures 1b and 1d). A reliability analysis to compare HH with BL using a two-tailed paired t-test indicates that the findings are reliably different across all prior hint history with p=0. There is a consistency of results for all performance metrics for HH, while BL exhibits more prominent fluctuation in its results, suggesting that the HH model can be feasibly used to predict student hint request behavior for both unseen skills and unseen students, as well as across the number of first action history points with fair reliability.

5. CONTRIBUTION AND FUTURE WORK

Results of the experiments suggest that students' help request behavior can be feasibly predicted from data that are descriptive of student action information. While the methods in this study are a starting point in using action information, we feel that such initiatives are worth discussing for building up further studies in the field. The models provide utility for predicting when students will ask for help, using dataset information on problem attempts and help requests. Both models predicted students' first course of action when answering problems from an ITS with fairly consistent predictive performance and generalizability.

Future improvements to these models may include the accounting of patterns in student actions which may provide a rich source of information for possible prediction of need for assistance by students (partly explored here with the BL model). The dataset used contained other information including student response times and skill difficulty and exploiting these may provide further insight into factors of assistance need to aid in developing a proactive and effective early intervention framework. These models should be tested on other ITS datasets to determine whether these models are consistent across different datasets.

REFERENCES

[1] Corbett, A.T. and Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction 4, 4 (1994), 253–278.

[2] Duong, H., Zhu, L., Wang, Y. and Heffernan, N.T. (2013). A prediction model that uses the sequence of attempts and hints to better predict knowledge: "Better to attempt the problem first, rather than ask for a hint". In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 316-317.

[3] Mathews, M. and Mitrović, T. How Does Students' Help-Seeking Behaviour Affect Learning? In B.P. Woolf, E. Aïmeur, R. Nkambou and S. Lajoie, eds., Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2008, 363–372.

[4] Pavlik, P. I., Cen, H., and Koedinger, K. (2009) Performance Factors Analysis - A New Alternative to Knowledge Tracing. Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK, pp. 531-538.

[5] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek, M.A., and Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) Proceedings of the 12th Artificial Intelligence in Education, Amsterdam: ISO Press. pp. 555-562.

[6] Wang, Y. and Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. The 24th International FLAIRS Conference.

[7] Wang, Q.Y., Kehrer, P., Pardos, Z. and Heffernan, N. Response Tabling– A simple and practical complement to Knowledge Tracing. KDD 2011 Workshop: Knowledge Discovery in Educational Data.

Modeling Students' Memory for Application in Adaptive Educational Systems

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

Human memory has been thoroughly studied and modeled in psychology, but mainly in laboratory setting under simplified conditions. For application in practical adaptive educational systems we need simple and robust models which can cope with aspects like varied prior knowledge or multiple-choice questions. We discuss and evaluate several models of this type. We show that using the extensive data sets collected by online educational systems it is possible to build well calibrated models and get interesting insight, which can be used for improvement of adaptive educational systems.

1. INTRODUCTION

Development of intelligent tutoring system and other adaptive educational systems is often focused on teaching mathematics, physics, and similar domains. The related research in student modeling is thus concerned mainly with modeling skill acquisition. Another interesting area, where adaptability is very useful, is learning of facts [8], particularly in domains with varied prior knowledge like vocabulary, geography, or human anatomy. In this context, modeling of students' memory is important.

Principles of human memory and their consequences for education have been extensively studied in psychology, e.g., [2, 5, 9, 10]. Models developed in the psychological research are not, however, easily applicable in practical implementation of adaptive practice. The purpose of models described in psychological literature is to describe and explain mechanisms of human memory, e.g., the spacing effect [9]. Experiments are done using lab studies under controlled setting, in areas with little prior knowledge, e.g., learning of arbitrary word lists, nonsense syllables, obscure facts, or Japanese vocabulary.

In the context of development of adaptive educational systems, our goal is more pragmatic – we do not need to capture all details of human memory, we need a model which will work well in an adaptive system. A model needs to provide good input for other modules of an adaptive system (e.g., question selection or open learner model). The specific context of our work is an adaptive application `slepemapy.cz` for learning geography [8].

Although we can afford to model memory in a simplified manner, we have to deal with issues like varied prior knowledge, multiple-choice questions (with possibility of guessing), and no control on when students use the system. Compared to laboratory studies online educational systems can easily collect much more extensive data (millions of answers), so we can employ machine learning techniques to find fitting models. Specifically, in our work we use this approach to detect the dependence of memory activation on time from previous answer. The standard approach [9] is to make an assumption about the functional form of such dependence. We learn the function from the data and it turns out to be an S-shaped function which cannot be represented symbolically in a straightforward way. The results also show that there are large differences between learning of facts even in a seemingly compact domain like geography. These results may be useful for improving the behaviour of adaptive educational systems.

2. MODELING

Before we go into the description of models, let us clarify the context of considered models. In previous work [8] we described a modular architecture for an adaptive practice of facts based on three modules: estimation of prior knowledge, estimation of current knowledge, construction of questions. Here we focus on improving the estimation of current knowledge by taking timing between answer into account.

Specifically, we assume the following input: for each student and repeatedly answered fact (e.g., a country in the case of our application), we have an initial estimate of the student's knowledge of the fact and data about a sequence of student's answers. For each answer we consider the correctness of the answer, the type of question (either open question or multiple-choice question with a specified number of options), and time from previous answer (in seconds). For estimating initial activation we use a variant of the Elo rating system [4, 13] as specified in [8]. For purpose of this work this estimation is treated as a black box.

As an output a model provides estimated probability that the next answer will be correct. This output can be used for the adaptive construction of questions (in such a way that

they have appropriate difficulty) [7, 8]. Model parameters can be also used for presenting feedback to students in the form of an open learner model.

2.1 Basic Approach

Student models of learning [3] most commonly use either a binary skill (a typical model of this type is Bayesian Knowledge Tracing) or a continuous skill with probability of correct answer specified by the logistic function of the skill. For modeling memory it is natural to use a continuous skill since memory is build gradually – as opposed, for example, to understanding or insight in mathematics, which may undergo sudden transition from unlearned to learned state as assumed by Bayesian Knowledge Tracing [1]. Modeling based on the logistic function was also previously used for modeling memory [9]. In the following we use the notion of *memory activation* instead of skill.

All models that we consider have the following basic form. Based on the data we estimate memory activation m. Probability that the next answer will be correct is estimated using a logistic function: $P(m) = \frac{1}{1+e^{-m}}$. In the case of multiple-choice question with n options the probability of correct answer is given by the shifted logistic function: $P(m) = \frac{1}{n} + (1 - \frac{1}{n})\frac{1}{1+e^{-m}}$. Note that this functional form is a simplification, since it does not consider the possibility that a student answers correctly by ruling out distractors.

2.2 Computing Memory Activation

A basic model applicable under the outlined approach is a simplified, one-dimensional variant of Performance Factor Analysis (PFA) [11] (originally PFA was formulated in terms of skills and vectors, as it uses multiple knowledge components). In this model the memory activation is given by a linear combination of an initial activation and past successes and failures of a student: $m = \beta + \gamma s + \delta f$, where β is the initial activation, s and f are counts of previous successes and failures of the student, γ and δ are parameters that determine the change of the skill associated with correct and incorrect answers. The basic disadvantage of this simple approach is that it does not consider the time between attempts; in fact it even ignores the order of answers (it uses only the summary number of correct and incorrect answers).

ACT-R model [9, 12] of spacing effects can be considered as an extension of this basic model. In this model the memory activation is estimated as $m = \beta + \log(\sum b_i t_i^{-d_i})$, where the sum is over all previous attempts, values t_i are the ages of previous attempts, values b_i capture the influence of correctness of answers, d_i is the decay rate, which is computed by recursive equations [9]. The model also includes additional modifiers for treating time between sessions. The focus of the model is on modeling the decay rate to capture the spacing effect. Studies using this model [9, 12] did not take into account the probability of guessing and variable initial knowledge of different items (initial activation was either a global constant or a student parameter). In the current work we focus on these factors and for the moment omit modeling of spacing effects.

Another possible extension [8] of the basic PFA model is to combine it with some aspects of the Elo rating system [4, 13]; in the following we denote this version as PFAE (PFA Elo/Extended). The estimated memory activation is updated after each answer as follows:

$$m := \begin{cases} m + \gamma \cdot (1 - P(m)) & \text{if the answer was correct} \\ m + \delta \cdot P(m) & \text{if the answer was incorrect} \end{cases}$$

To include the timing information into this model, we can locally increase the memory activation for the purpose of prediction, i.e., instead of $P(m)$ to use $P(m + f(t))$, where t is the time (in seconds) from the last attempt and f is a *time effect function*. As m denotes memory activation, the value $f(t)$ corresponds to temporal increase in memory activation due to (short) time from previous exposure of an item.

It is natural to use as a time effect function some simple analytic function, but analysis of our data suggests that this approach does not work well. Figure 1 shows calibration analysis for two time effect functions: $f(t) = \frac{w}{t}$ (used in previous work [8]) and $f(t) = 1.6 - 0.1 \log(t)$ (the functional form is based on [9] and fitted to data). We see that neither of these functions leads to well calibrated predictions. Since we were not able to find a simple time effect function that would provide a good fit, we represent the function $f(t)$ as a staircase function with fixed bounds $\vec{b}$ and values $\vec{v}$ which we learn from the data:

$$f(t) = \begin{cases} v_i & \text{if } b_i \leq t < b_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

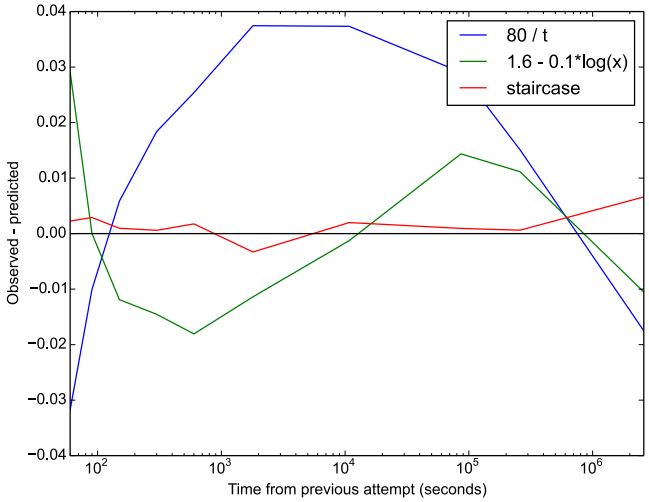


Figure 1: Calibration for the PFAE model with different time effect functions – the y axis shows difference between observed frequency of correct answers and average prediction.

3. EXPERIMENTS

We report experiments with the PFAE model with time effect function. For evaluation we used data from an online system for practicing geography [8] (`slepemapy.cz`). Data were filtered to include only students with at least 20 answers, items (places) with at least 40 answers, and we consider only sequences where a student answered at least 3 questions about an item. For experiments we divided the data into 10 sets, each containing 52,190 sequences of answers.

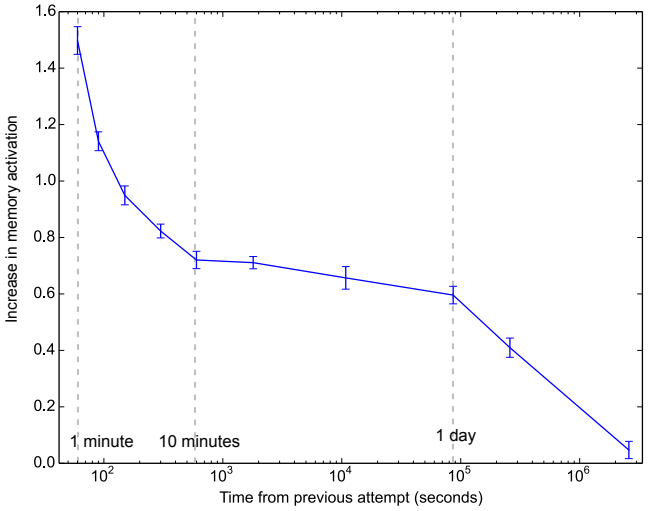


Figure 2: Time effect function – average from 10 independent data sets, error bars show standard deviations of parameter estimates.

3.1 Model Parameters

As the fixed bounds used in the staircase representation of time effect function we have chosen the following values: 0, 60, 90, 150, 300, 600, 1800, 10800, 86400, 259200, 2592000. These values were chosen to be easily interpretable (e.g., 30 minutes, 1 day) and at the same time to have reasonably even distribution of data into individual bins.

The model has the following parameters which we need to estimate from the data: update constants γ, δ and the vector $\vec{v}$ representing the time effect function. To estimate these parameters we use a gradient descent. To evaluate stability of parameter estimates we computed the parameter values for the 10 independent data sets. The results show that the obtained parameters are very stable: $\gamma = 2.290 \pm 0.042$, $\delta = -0.917 \pm 0.018$; values $\vec{v}$ for the representation of time effect function are depicted in Figure 2.

Since our data set is large and parameter estimates are stable, we can afford to do more detailed analysis. Figure 3 shows fitted time effect functions and γ, δ values when the parameters are fitted using only part of the data. Figure 3 A shows that there is quite large difference between parameter values for cases with high and low prior knowledge. This suggests possible improvement to the PFAE model – not just by including more parameters, but also by changing its functional form. However, prior knowledge is not the only factor that plays role. Figure 3 B shows fitted parameters for several types of places. In all of these cases the prior knowledge is low, yet there are still large differences between fitted parameters values. These parameters may contain useful information about students' learning in particular parts of the domain, e.g., data in Figure 3 B illustrate that it is easier to learn states of Germany than provinces of China.

In the case of countries we have enough data to perform parameter fitting for individual places. In this case we fix the time effect function (as learned on the whole data set and reported in Figure 2) and we learn only the γ, δ parameters on

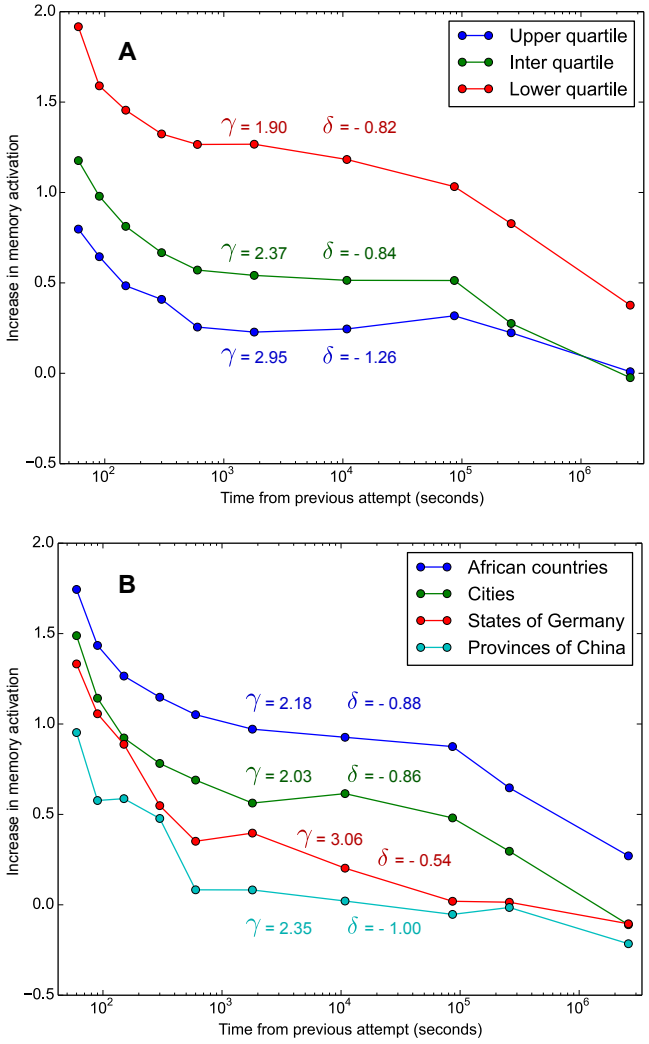


Figure 3: Time effect function and γ, δ parameters fitted to filtered data: A) by estimated prior knowledge, B) by the type of a place.

data for a single place. We use only places for which we have at least 1300 students answering at least 3 questions. The fitted parameter γ is has an interpretable meaning "how easy it is to remember a country". Examples of countries with high γ (>3.3): Western Sahara, Southern Sudan, Vietnam, Egypt, Somalia; countries with low γ (< 1.7): Bulgaria, Romania, Serbia, Moldova. Note that the reported results are clearly dependent on the origin of students using the system – in our case mostly Czech students.

3.2 Accuracy of Predictions

Table 1 show comparison of several model variants with respect to three common performance metrics [14]: root mean square error (RMSE), log-likelihood (LL), and area under the ROC curve (AUC). The results show averages from 10 runs on different training/testing sets. The results are consistent over the three metrics and show that the PFAE models brings quite large improvement over the PFA model. Differences between variants of the PFAE model due to the used time effect function are statistically significant, but other-

Table 1: Comparison of models with respect to three performance metrics.

model	time effect	RMSE	LL	AUC
PFA	–	0.3593	-106517	0.719
PFA	$80/t$	0.353	-103441	0.7195
PFAE	$80/t$	0.3377	-94454	0.757
PFAE	$1.6 - 0.1\log(t)$	0.3367	-93987	0.7591
PFAE	staircase	0.3363	-93642	0.7614

wise rather small. Individual predictions are actually highly correlated (correlation coefficient around 0.97).

4. DISCUSSION

We have evaluated several variants of a model of memory activation in the context of adaptive practice of facts. We proposed a model which incorporates the effect of time from previous answer by a general staircase function, which is learned from data (as opposed to assuming a specific symbolic form of the function). The model is better calibrated than other studied models and provides slightly better predictions. More importantly, the model is simple, parameters are easy to learn from data and robust. The learned function also provides interesting insight into students memory in the particular application – there is fast decrease in memory activation within the first 10 minutes, then the effect is nearly steady for 1 day, after that the activation decreases again.

By performing fine-grained analysis of the data, it is possible to use the model parameters to determine items that are easy or difficult to remember. Such results may be useful for improvement of educational systems, e.g., by offering mnemonics for difficult to remember facts, or by changing the adaptive selection of questions to prefer easy to remember facts at the beginning of a session. Specifically, results reported in Figure 3 suggest that different adaptive behaviour may be useful for learning African countries and provinces of China.

A possible limitation of this study is that the used data do not come from a properly designed and controlled experiment, but from an adaptive system which uses a student model to choose questions [8]. This may potentially cause a bias in the performed analysis. Although it seems unlikely that the reported results would be significantly influenced by this data source, feedback loops between student models and data collection deserve attention [6].

Another simplification of the current work is that we do not consider the feedback provided by the used system when a student answers incorrectly. This feedback clearly has impact on memory activation of the selected wrong answer. This raises a more general question: What is more important for the practical development of adaptive educational systems – proper treatment of principal issues (e.g., spacing effect) or incorporation of practical features into the model (e.g., effect of wrong answers)?

Acknowledgement

The author thanks Vít Stanislav and Jan Papoušek for their work on the `slepemapy.cz` project and for their assistance with the data.

5. REFERENCES

[1] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[2] Peter F Delaney, Peter PJL Verkoeijen, and Arie Spirgel. Spacing and testing effects: A deeply critical, lengthy, and at times discursive review of the literature. *Psychology of learning and motivation*, 53:63–147, 2010.

[3] Michel C Desmarais and Ryan SJ d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[4] Arpad E Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[5] Jeffrey D Karpicke and Henry L Roediger. Repeated retrieval during learning is the key to long-term retention. *Journal of Memory and Language*, 57(2):151–162, 2007.

[6] Juraj Nižnan, Jan Papoušek, and Radek Pelánek. Exploring the role of small differences in predictive accuracy using simulated data. 2015. Submitted.

[7] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Proc. of Artificial Intelligence in Education (AIED)*, 2015.

[8] Jan Papoušek, Radek Pelánek, and Vít Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining (EDM)*, pages 6–13, 2014.

[9] Philip I Pavlik and John R Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29(4):559–586, 2005.

[10] Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

[11] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[12] Philip Pavlik Jr, Thomas Bolster, Sue-Mei Wu, Ken Koedinger, and Brian Macwhinney. Using optimally selected drill practice to train basic facts. In *Intelligent Tutoring Systems*, pages 593–602. Springer, 2008.

[13] Radek Pelánek. Time decay functions and Elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.

[14] Radek Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 2015. To appear.

Social Facilitation Effects by Pedagogical Conversational Agent: Lexical Network Analysis in an Online Explanation Task

Yugo Hayashi
Ritsumeikan University
56-1 Kitamachi, Toji-in, Kita-ku
Kyoto, 603-8577, Japan

yhayashi@fc.ritsumei.ac.jp

ABSTRACT

The present study investigates web-based learning activities of undergraduate students who generate explanations about a key concept taught in a large-scale classroom. The present study used an online system with Pedagogical Conversational Agent (PCA), asked to explain about the key concept from different points and provided suggestions and requests about how to make explanations, and gave social facilitation prompts such as providing examples by other members in the classroom. A total of 314 learner's text based explanation activities were collected from three different classrooms and were analyzed using the social network analysis methods. The main results from the lexical analysis show that those using the PCAs with social feedback worked harder to use more various types of explanations than those without such feedback. Future directions on how to design online tutoring systems are discussed.

Keywords

Online tutoring; Explanation activities; Social Facilitation; Lexical Network Analysis.

1. INTRODUCTION

Studies on designing intelligent tutoring systems, such as Pedagogical Conversational Agents (PCAs), which autonomously engage in learning activities, have suggested its effective use for learning, much like a human tutor [12, 9, 1]. Still, few studies empirically investigate the use of such technology for large numbers of students in a class and investigate the learner's cognitive processes. The present study investigated the unique designs of the user interface for learners that use an online tutoring system guided by a PCA in three different types of classes. The study especially focused on the use of PCAs in a concept-explanation activity task, where the PCA asked several questions for explanation and provided feedback such as social information about other members who were engaging in the task. We focused on how such feedback can increase the learner's explanation behaviors during such activities.

1.1 Facilitating explanation activities using PCAs

Studies on collaborative problem solving in the field of cognitive science reveal how concepts are understood or learned [3, 5]. Studies have shown that asking reflective questions for clarification to conversational partners is an effective interactional strategy to gain a deeper understanding of a problem or a concept [15, 16]. It has also been demonstrated that the use of strategic utterances, such as asking for explanation or providing suggestions, can stimulate reflective thinking and meta-cognition involved in understanding a concept. Based on these theories, there have been many attempts in the learning sciences to use such methods in classrooms [17, 13]. However, in an actual pedagogical situation, as in a large classroom, it is often difficult for one teacher to monitor learners and supervise their explanations. Recent studies [2, 11] have shown that the use of conversational agents that act as educational companions or tutors can facilitate learning process. Study [10] have shown that using PCAs that provide suggestions about how to make effective explanations can facilitate better motivation and improve task performance. Moreover, in a series of studies by the author, it is shown that the use of PCAs can provide affective feedback and facilitate better outcomes [7, 8, 6]. More specifically, the results show that PCAs with positive emotion motivates the learners to work harder compared to those without any emotional expressions. In this report, the author further investigated the effects of using such PCAs in an online explanation task. The study focused on a classroom of more than one hundred students who were using an online explanation task, where individuals made explanations to the PCA on a one-on-one basis, as an after school work activity. In such activity, the PCA will play the role of questioner and ask the student to explain about the key concept. The learners were students enrolled in a psychology class where their task was to make explanations about a key concept taught in their class, as an after class exercise.

1.2 Using social facilitating effects

One of the important factors that strongly influence human behavior in groups is the effect of the social influence produced by other members. Studies in social psychology have suggested that work efficiency is improved when someone is watching a person, i.e., the presence of an audience facilitates the performance of a task. The impact that an audience has on a task-performing participant is called the "audience effect." Another relevant concept on task efficiency is called "social facilitation theory" [19]. The theory claims that people tend to do better on a task when they are doing it in the presence of other people in a social situation; it implies that personal factors can make people more aware of social evaluation.

Coming back to the present study, even though the students made explanations about a concept to the PCA in a one-on-one situation, it was extremely important that they were aware that they were working in a social situation. Studies in media-psychology have provided much evidence that people lack social awareness in computer-mediated communication, compared to face-to-face communication [4]. Thus, it is effective to give information about the awareness of other learners online and create social

facilitations to make the learners become more active. One of the strong points of using online learning environments is that they are able to collect a huge amount of data from learners. A large database of dialogues of explanation texts may be reused for prompting hints or giving examples to learners who make explanations. It is also effective to provide information about the members who are working on the explanation task in real time or non-real time. If such kinds of feedback are used in online tutoring systems, it may facilitate learners' social awareness, and motivate their explanation activities.

Given all this, the present study investigated the effects of PCAs, which provide information about "other members", along with suggestions and comments about their explanations. The goal of the study is to investigate the how the quality of the learners explanations may change due to the facilitations from a PCA which encourages them to actively explain about key terms that were taught in class. The present study will use social network analysis method to capture the dynamics of diverse explanations during the online task. Unlike standard text analysis methods calculating the frequency of single important key terms that appear in the text, this method enables to detect different key terms that appear simultaneously in one explanation made by the learner. If the learner meets the expectations from the PCA, where it asks the learner to explain the key from various perspectives, different types of key terms should be used during their activity.

2. Method

The study was conducted in three large classes, each consisting of more than hundred students. We constructed an online web system that let learners make text-based explanations about key concepts taught in a psychology class. Students in an undergraduate psychology class used the system, and participated as part of their homework. A total of 30 different key terms (e.g., Gestalt, long-term memory, cognitive dissonance) were selected from the class and randomly assigned to each of the learners based on their IDs. On using the system, they were guided by a PCA that (1) instructed them on what to explain, (2) provided meta-cognitive suggestions, and (3) gave examples about how other members in the classroom made explanations.

2.1 Tutoring system for the experiment

A web-based tutoring system was developed only for the experiment using a web server, a database, and rule-based scripts. It was managed as a member-only system, and learners were required to login to the system for use. As mentioned in the previous section, each student was assigned to work on one randomly selected key term. As they logged into the system, a PCA appeared on the screen and stated the selected key concept, and gave him/her questions about how to explain it. The task was comprised by 17 trials with two major steps in each trial as follows: (a) text-input and, (b) feedback from the PCA.

On the first (Trial 1) and the final trials (Trial 17) the PCA asked the learner to input freely regarding whatever they knew about the key concept. These are taken as pre- and post- tests were they can freely input the messages as a free recall test. Through the 2nd and 16th trials, the learners were given specific questions about what to explain about the keyword. For example, the PCA may ask a series of question such as "How can it be used", "What is it similar to", or "In what period of time you use it" etc. These trials are considered as the explanation/training phase. The PCA also encourages the learner to think on their own way and input individual unique explanations.

On each trial, they were asked to do the following: (1) input explanations and click on the next button, (2) read the provided meta-suggestions from the PCA to make effective explanations, and depending on the experimental condition (explained in the next section), it provided information about other members who also responded for the given key concept.

To facilitate the social presence of the other members and make learners to think in their own way, the study uses two types of prompts. First, the utterances of other learners who had already inputted into the system were used. These messages were presented along with the initials of the person who answered the explanation. This enabled them to be aware how many in the class were working on the same key term. The utterances of other group members were only shown after the learner inputted his/her answers, and so the leaner couldn't simply copy and paste other's explanations during their trial.

2.2 Experiment design and learners

The experiment was conducted in three classes where each class was assigned to an experimental condition. In one class (the baseline condition), all learners were assigned to use PCAs without any social awareness functions or examples of other learners. The PCA only provided back-channel feedback and gave meta-suggestions about how to make explanations more effectively (e.g., Try to think from various viewpoints). These suggestions were compiled from a previous study [7]. In another class (the example condition), the learners were assigned to use the PCAs with additional functions, which provided examples of answers inputted by other members. The third class (the example+ condition) was assigned to those in the example condition with PCAs with additional functions. In other words, they were presented with examples with explanations of others, plus information about the number of members who were assigned to work on that key concept. There were 105 Japanese undergraduates (55 males, 50 females, mean age = 18.26 years) in the baseline condition. In the example condition, there were 105 Japanese undergraduates (55 males, 50 females, mean age = 18.46 years). Finally, in the example+ condition, there were 104 undergraduates (52 males, 52 females, mean age = 18.35 years).

3. RESULTS

3.1 Lexical Network Analysis

The text analysis was comprised by several steps such as (1) morphologically analyzing the text data, (2) developing a dictionary database using a thesaurus, and (3) conducting lexical network analysis to understand the usage of variety of different words during their final explanation. Recently, such social network analysis method is adopted to investigate the usage of important words in collaborative learning [8, 16].

3.1.1 Preprocessing

The recorded texts were broken down into morphemes with the Japanese morphological analysis tool MeCab (Java Sen port: *http://mecab.sourceforge.net* (accessed April 2015)). The objective of the first stage of the analysis was to extract the most frequent morphemes, such as the nouns and verbs through all learners textual inputs. 105,488 morphemes were collected and the most 28 frequent words were chosen as important words for explanations. Those were labeled based on the thesaurus

dictionary database such as: 'presence', 'causal', 'relations', 'actions', 'thought', 'matters', 'case', 'conclude', 'understand', 'analogy', 'predict', 'logic', 'reason', 'hypothesis', 'convergence', 'explanation', 'intention', 'theory', 'relative', 'knowledge', 'explicate', 'transform', 'opposition', 'compliment', 'compare', 'inevitability', 'method', and 'reason' [14].

Additionally, based on the semantic hierarchical structure of the thesaurus, new keywords were added to the dictionary database that were related to the 28 keywords. This was done to capture all the semantically related words to these keywords. As a result, 2,722 new words that have relative meanings to the keywords were registered into the semantic dictionary database.

3.1.2 Network Analysis

Using the semantic dictionary database as training data set, the learners textual inputs were further analyzed. For each trial input, the number of appearing semantic keywords in the dictionary were counted. The data of these semantic key words were then analyzed by adopting the social network analysis method. This method was used to analyze the co-occurrence between keywords, i.e. *capturing the diversity of the types of words that were used during one explanation*. The network was developed based on a bipartite graph of keywords x explanations(trials). Since the PCA provided various questions and enforced them to explain uniquely along with their social feedbacks during their explanation activities(trial 2 to 16), their achievements should be reflected to their explanation activities. Learners should use more different types of key terms in the example+ condition since they are facilitated more strongly to take different perspectives by mentioning about other group members presence. Each node in a network was represented as the semantic category of the keyword that was frequently used during their explanation. The threshold of a node(semantic keyword) determining as frequently used or not was defined based on the comparison by the average of other nodes. The threshold of a node n was determined as follows:

$$\theta = \begin{cases} 1(n \geq \bar{n}) \\ 0(n < \bar{n}) \end{cases} \quad (1)$$

On investigating the differences between conditions and over time, the number of links connecting each nodes were calculated. The following equation represents the amount of density where n stands for the number of nodes and l stands for the number of links:

$$d = \frac{l}{n(n-1)} \quad (2)$$

Table 1 shows the quantitative results of the lexical network analysis. The results suggest that at the pre-test (1st trial), learners had only few connections between nodes, thus indicating that the variations of words were few in terms of semantic categories. On the post-test (17th trial), the connections of nodes increased due to conditions. This shows that learners used more variety of words during explanations in the post-test(17th trial) example+ condition(0.27) than example(0.24) and baseline(0.15) conditions. The results gives us a clear vision of the dynamics of explanations they gave to the agent differ due to the conditions using more social awareness designs.

Table 1. The score of density of each conditions performed by the lexical network analysis.

Conditions	Pre (1st trial)	Post (17th trial)
baseline	0.07	0.15
example	0.06	0.24
example+	0.06	0.27

The analysis above shows that learners were using more different key terms at the same time in each trial. However it lacks in evidence rather if they tried to use different key terms in their post test compared from those in the pre-test. They might have simply used the same words they inputted from their first trial. It is it is important in this learning context that to know if they changed their phrases or tried to use more sophisticated words from the initial state of the explanation activity. Therefore, additional analysis was conducted to investigating the network similarity between the pre(1st) and post(17th) trial. The following correlation index was adopted on calculating the similarity between the two networks.

$$c = \frac{\sum_{i=1}^{784}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{784}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{784}(b_i - \bar{b})^2}} \quad (3)$$

a and b stands for the number of nodes in the bipartite graph each pre- and post-test respectively. Figure 1 indicates the results of c for each condition.

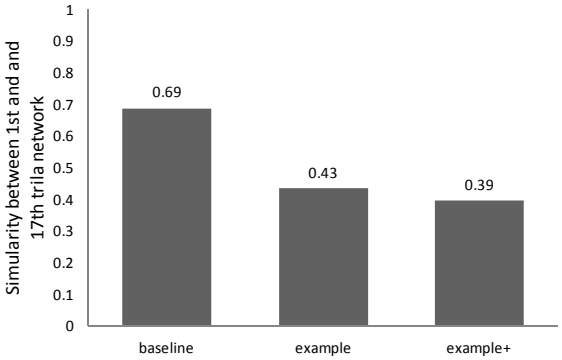


Figure 1. Results of similarity between the pre(1st) and post(17th) trial in each condition

The results indicate that learners in the baseline condition used more similar words from the pre-test on their final post-test explanations(0.69). On the other hand, learners in the example+ and example condition shows that they were using more different key terms compared to those from those in the 1st trial(0.43, 0.39 respectively).

The analysis from the series of analysis indicates that learners with social facilitation (1) used more different key terms simultaneously in their final explanation activities, and (2) those were different from those in the initial explanation activities. This analysis captures a new view from the study of [8] where it did not investigate the changes of the network over time.

4. DISCUSSION AND CONCLUSION

The present study investigated the use of PCAs in an online explanation activity where students were required to make explanations about a key concept. The focus here was to investigate the effects of social facilitations over time, using a large scale database collected during the online explanation task. These social facilitations were provided through a PCA during the learner's explanation activities and they were to enhance the co-presence of other classmates and motivate their activities by encouraging them. In the experiment, students enrolled in three psychology classes used an online explanation system and made explanations to the PCA. They also received comments on how to make effective explanations along with social feedbacks of other classmates. The results of the text analysis show that learners tend to input more important messages simultaneously in the final trial compared to the first trial when they received feedback about other group members (example and example+ condition). This indicates that this type of social feedback can motivate learners to work harder and facilitate effective explanation over time. An interesting point is that even though all the students were told that their answers would not be graded, they still tried harder when they were shown some of the other members' activities. This shows that the effects of the "audience" and "social facilitation" are quite strong in such situations. The results can be interpreted that the situation given to the learner are useful to make the learners aware that their messages could be seen by other in-group members and thus this might have made them work harder in their activities. Another interpretation is that showing others' comments might have allowed learners to avoid negative feelings and thoughts, such as he/she might have inputted something very out of line. As explained earlier in this paper, novice learners have difficulty making explanations to others [5]. Thus, it may be assumed that learners in the baseline condition experienced negative feelings, worrying that they were making mistakes about the text. On the other hand, the use of the examples and the social contexts in the example and example+ conditions may have eased such negative feelings, and thus, increased self-confidence compared to the baseline condition. This study provided implications about how to design effective online tutoring systems, incorporating PCAs with information about other working members, thus providing social facilitation.

5. ACKNOWLEDGMENTS

This work was supported (in part) by 2012 KDDI Foundation Research Grant Program and the Grant-in-Aid for Scientific Research (KAKENHI), The Ministry of Education, Culture, Sports, Science, and Technology, Japan (MEXTGrant), Grant No. 25870910.

6. REFERENCES

[1] Baylor, A.L. and Kim, Y. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education 15*, 1 (2005), 95-115.

[2] Baylor, A.L. and Ryu, J. The API (Agent Persona Instrument) for Assessing Pedagogical Agent Persona. In *Proc. World Conference on Educational Multimedia, Hypermedia and Telecommunications*, D. Lassner and C. McNaught (eds.) (2003), 448-451.

[3] Chi, M., Leeuw, N., Chiu, M. and Lavancher, C. Eliciting self-explanations improves understanding. *Cognitive Science 18*, 3, (1994), 439-477.

[4] Clark, H.H. and Brennan, S.E. Grounding in communication. In B. L. Resnick, M. R. Levine, and D. S. Teasley (Eds.), *Perspectives on socially shared cognition*, APA Press, 1991, 127-149.

[5] Graesser, A. and McNamara, D. Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist 45*, 4 (2010), 234-244.

[6] Hayashi, Y. Learner-support agents for collaborative interaction: A study on affect and communication channels. In *Proc. 10th International Conference on Computer Supported Collaborative Learning* (2013), 232-239.

[7] Hayashi, Y. On pedagogical effects of learner support agents in collaborative interaction. In *Proc. 11th International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science* (Cerri, A. S. and Clancey, B. eds.), Springer-Verlag, 7315 (2012), 22-32.

[8] Hayashi, Y. Explanation activities with a pedagogical agent in an online task: Lexical network analysis. In *Proc. CHI 2015 Works-in-Progress*, 1457-1460.

[9] Heidig, S. and Clarebout, G. Do pedagogical agents make a difference to student motivation and learning? *Educational Research Review 6*, 1 (2011), 27-54.

[10] Holmes, J. Designing agents to support learning by explaining. *Computers & Education 48*, 4 (2007), 523-547.

[11] Kim, Y., Baylor, A.L. and Shen, E. Pedagogical agents as learning companions: The impact of agent emotion and gender. *Journal of Computer Assisted Learning 23*, 3 (2007), 220-234.

[12] Kumar, R. and Rose, C. Architecture for building conversational agents that support collaborative learning. *IEEE Transactions on Learning Technologies 4*, 1 (2011), 21-34.

[13] Miyake, N. Constructive interaction and the interactive process of understanding. *Cognitive Science 10*, (1986), 151-177.

[14] National Institute for Japanese Language and Linguistics, *Bunruigoihyou*, Dainippon publications, (2004). (In Japanese)

[15] Okada, T. and Simon, H. Collaborative discovery in a scientific domain. *Cognitive Science 21*, 2 (1997), 109-146, 1997.

[16] Oshima, J., Matsuzawa, Y., Oshima, R., Chan, C. K. K., & van Aalst, J. Social Network Analysis for Knowledge Building: Establishment of Indicators for Collective Knowledge Advancement. In J. van Aalst, K. Thompson, M. J. Jacobson, & P. Reinmann (Eds.), *The Future of Learning: Proceedings of the 10th International Conference of the Learning Sciences (ICLS2012)*, 2 (2012), 465-466.

[17] Salomon, G. *Distributed Cognition: Psychological and Educational Considerations*. Cambridge University Press, New York, USA, 2001.

[18] Shirouzu, H., Miyake, N. and Masukawa, H. Cognitively active externalization for situated reflection. *Cognitive Science 26*, 4 (2002), 469-501.

[19] Zajonc, R.B. Social facilitation. *Science 149*, (1965) 271-274.

Personalized Education; Solving a Group Formation and Scheduling Problem for Educational Content

Sanaz Bahargam, Dóra Erdős, Azer Bestavros, Evimaria Terzi
Computer Science Department, Boston University, Boston MA
[bahargam, edori, best, evimaria]@cs.bu.edu

ABSTRACT

Wether teaching in a classroom or a Massive Online Open Course it is crucial to present the material in a way that benefits the audience as a *whole*. We identify two important tasks to solve towards this objective; (1.) group students so that they can maximally benefit from peer interaction and (2.) find an optimal schedule of the educational material for each group. Thus, in this paper we solve the problem of team formation and content scheduling for education. Given a time frame d, a set of students $\mathbf{S}$ with their required need to learn different activities $\mathbf{T}$ and given k as the number of desired groups, we study the problem of finding k group of students. The goal is to teach students within time frame d such that their potential for learning is maximized and find the best schedule for each group. We show this problem to be NP-hard and develop a polynomial algorithm for it. We show our algorithm to be effective both on synthetic as well as a real data set. For our experiments we use real data on students' grades in a Computer Science department. As part of our contribution we release a semi-synthetic dataset that mimics the properties of the real data.

Keywords

Team Formation; Clustering; Partitioning; Teams; MOOC

1. INTRODUCTION

Many work has been dedicated on how to improve students' learning outcome. We recognize two substantial conclusions; first, the use of personalized education. By shaping the content and delivery of the lessons to the individual ability and need of each student we can enhance their performance([6, 11, 12]. Second, grouping students; working in teams with their peers helps students to access the material from a different viewpoint as well [7, 4, 13, 1]. In this paper we study the problem of creating personalized educational material for teams of students by taking a computational perspective. To the best of our knowledge we are the first to formally define and study the two problems of team formation and personalized scheduling for teams in the context of education. We present a formal definition for these problems, study their computational complexity and design algorithms for solving them. In addition, we also apply our algorithms to a real dataset obtained from real students. We make our semi-synthetic dataset `BUCSSynth`, generated to faithfully mimic the real student data available on our website.

Related Work: Besides the work on improving students learning outcome, related problems have also been studied in computer science. Topics of interest are team formation [2, 3, 9, 10] and scheduling theory, see [5] for an overview.

2. PRELIMINARIES

We model a student's learning process by a sequence of topics that she learns about. In this sequence topics may appear multiple times, and repetitions of a topic may count with different weights towards the overall benefit of the student. Let $\mathbf{S} = \{s_1, s_2, \ldots, s_n\}$ be a set of students and $\mathbf{T} = \{t_1, t_2, \ldots, t_m\}$ be a set of topics. We assign topics to d timeslots,a *schedule* $\mathcal{A}$ is a collision free assignment of topics to the timeslots. $\mathcal{A}$ can be thought of as an ordered list of (possible multiple occurrences) of the topics. For a topic $t \in \mathbf{T}$ the tuple $\langle t, i\rangle$ denotes the i^{th} occurrence of t in a schedule. The notation $\mathcal{A}[r] = \langle t, i\rangle$ refers to the tuple $\langle t, i\rangle$ that is assigned to timeslot r in $\mathcal{A}$.

For student $s \in \mathbf{S}$ and topic $t \in \mathbf{T}$ the *requirement* $\texttt{req}(s, t)$ is an integer depicting the number of times s needs to learn about t to master its content. We assume that for the first $\texttt{req}(s, t)$ repetitions of t there is some benefit to s from every repetition of t, but for any further repetition there is no additional benefit to s. We call $\mathbf{b}(s, \langle t, i\rangle)$ (Equation (1)) the *benefit* of s from hearing about t for the i^{th} time.

$$\mathbf{b}(s, \langle t, i\rangle) = \begin{cases} \frac{1}{\texttt{req}(s.t)} & \text{if } i \leq \texttt{req}(s, t) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Note that for ease of exposition, we assume that all repetitions of t before $\texttt{req}(s, t)$ carry equal benefit to s. However, the definition and all of our later algorithms could easily be extended to use some other function $\mathbf{b}'(s, \langle t, i\rangle)$.

Given the benefits $\mathbf{b}(s, \langle t, i\rangle)$ there is a natural extension to define the benefit $\mathbf{B}(s, \mathcal{A})$ that s gains from schedule $\mathcal{A}$. This benefit is simply a summation over all timeslots in $\mathcal{A}$,

$$\mathbf{B}(s, \mathcal{A}) = \sum_{r=1}^{d} \mathbf{b}(s, \mathcal{A}[r]) \quad (2)$$

3. THE GROUP SCHEDULE PROBLEM

Given a group of students $P \subseteq \mathbf{S}$ our first task is to find an optimal schedule for P. That is, find a schedule to maximize the *group benefit* $\mathbf{B}(P, \mathcal{A})$ that group P has from $\mathcal{A}$ (Equation (3)).

$$\mathbf{B}(P, \mathcal{A}) = \sum_{s \in P} \sum_{r=1}^{d} \mathbf{b}(s, \mathcal{A}[r]) \tag{3}$$

We call this the GROUP SCHEDULE problem (problem 1).

PROBLEM 1 (GROUP SCHEDULE). *Let $P \subseteq \mathbf{S}$ be a group of students and $\mathbf{T}$ be a set of topics. For every $s \in \mathbf{S}$ and $t \in \mathbf{T}$ let $\texttt{req}(s,t)$ be the requirement of s on t given for every student-topic pair. Find a schedule $\mathcal{A}_P$, such that $\mathbf{B}(P, \mathcal{A}_P)$ is maximized for a deadline d.*

The `Schedule` algorithm. We first give a simple polynomial time algorithm, $\texttt{Schedule}(P, d)$ (Algorithm 1), to solve problem 1. `Schedule` is a greedy algorithm that assigns to every timeslot an instance of the topic with the largest marginal benefit. We say that the *marginal benefit*, $\mathbf{m}(P, \langle t, i \rangle)$, from the i^{th} repetition of t (thus $\langle t, i \rangle$) to P is the increase in the group benefit if $\langle t, i \rangle$ is added to $\mathcal{A}$. The marginal benefit can be computed as the sum of benefits over all students in P as given in Equation (4).

$$\mathbf{m}(P, \langle t, i \rangle) = \sum_{s \in P} \mathbf{b}(s, \langle t, i \rangle) \tag{4}$$

The `Schedule` algorithm is an iterative algorithm with d iterations that in every iteration appends a topic to the schedule $\mathcal{A}_P$. We maintain an array B in which values are marginal benefit of topics t, and an array R that contains a counter for every topic in $\mathcal{A}_P$. In every iteration `Schedule` selects the topic u_t with the largest marginal benefit from B and adds it to $\mathcal{A}_P$ (Lines 5 and 6) . Then it updates marginal benefit of u_t, $B[u_t]$ (Lines 7- 8). It is easy to see that Algorithm 1 yields an optimal schedule for a group P and runs in $O(d(|P| + log|\mathbf{T}|))$.

Algorithm 1 `Schedule` algorithm for computing an optimal schedule $\mathcal{A}_P$ for a group P.

Input: requirements $\texttt{req}(s,t)$ for every $s \in P$ and every topic $t \in \mathbf{T}$, deadline d.
Output: schedule $\mathcal{A}_P$.

1: $\mathcal{A}_P \leftarrow [\,]$
2: $B \leftarrow [\mathbf{m}(P, \langle t, 1 \rangle)]$ for $t \in \mathbf{T}$
3: $R \leftarrow [0]$ for all $t \in \mathbf{T}$
4: **while** $|\mathcal{A}_P| < d$ **do**
5: Find topic u_t with maximum marginal benefit in B
6: $\mathcal{A}_P \leftarrow \langle u_t, R[u_t] \rangle$
7: $R[u_t] + +$
8: Update $B[u_t]$ to $\mathbf{m}(P, \langle t, R[u_t] \rangle)$
9: **end while**

4. THE COHORT SELECTION PROBLEM

The next natural question is, that given a certain teaching capacity K (i.e., there are K teachers or K classrooms available), how to divide students into K groups so that each student benefits the most possible from this arrangement. At a high level we solve an instance of a partition problem; find a K-part partition $\mathcal{P} = P_1 \cup^* P_2 \cup^* \ldots \cup^* P_K$ of students into groups, so that the sum of the group benefits over all groups is maximized. This is the COHORT SELECTION Problem .

PROBLEM 2 (COHORT SELECTION). *Let $\mathbf{S}$ be a set of students and $\mathbf{T}$ be a set of topics. For every $s \in \mathbf{S}$ and $t \in \mathbf{T}$ let $\texttt{req}(s,t)$ be the requirement of s on t that is given. Find a partition $\mathcal{P}$ of students into K groups, such that*

$$\mathbf{B}(\mathcal{P}, d) = \sum_{P \in \mathcal{P}} \mathbf{B}(P, \mathcal{A}_P) \tag{5}$$

is maximized, where $\mathcal{A}_P = \texttt{Schedule}(P, d)$ for every group.

The COHORT SELECTION (Problem 2) is NP-hard as the Catalog Segmentation problem [8] can be reduced to it.

4.1 Partition algorithms.

In this section we introduce `CohPart` (Algorithm 3) as our solution to the COHORT SELECTION problem. The input to Algorithm 3 are the requirements $\texttt{req}(s,t)$, number of groups K and length of the schedule d. The output is a partition $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$ of the students and corresponding schedules $\{\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_K\}$ for each group.

`CohPart` first assigns every student to one of the groups in $\mathcal{P}$ at random (Line 3) and an initial optimal schedule for every group is computed (Line 5). Then in every iteration of the algorithm first every student is assigned to the group with the highest benefit schedule for the student (Line 9) and then the group schedules are recomputed (Line 12). The runtime of each iteration is $O(k|\mathbf{S}||\mathbf{T}|)$. In our experiments we observed that our algorithm converges really fast, less than a few tens of iterations.

Algorithm 2 `Benefit` algorithm to compute the benefit for student s from schedule $\mathcal{A}$

Input: requirements $\texttt{req}(s,t)$ for a student $s \in P$ and every topic $t \in \mathbf{T}$ and a single schedule $\mathcal{A}$
Output: $\texttt{Benefit}(s, \mathcal{A})$ Benefit of s from schedule $\mathcal{A}$.

1: $\texttt{Benefit}(s, \mathcal{A}) = 0$
2: **for** all topics $t \in \mathbf{T}$ **do**
3: $\texttt{Benefit}(s, \mathcal{A}) = \texttt{Benefit}(s, \mathcal{A}) + \frac{min(req(s,t), \mathcal{A}[t])}{\mathcal{A}[t]}$
4: **end for**

5. EXPERIMENTS

The goal of these experiments is to gain an understanding of how our clustering algorithm works in terms of performance (objective function) and runtime. Furthermore, we want to understand how the deadline parameter impacts our algorithm. We used a real world dataset, semi synthetic and synthetic datasets. The semi synthetic dataset and the source code to generate it are available in our website. [1] We first explain different datasets and then show how well our algorithm is doing on each dataset.

5.1 Algorithms

We compare `CohPart` to two baseline algorithms.

[1] `http://cs-people.bu.edu/bahargam/edm/`

Algorithm 3 `CohPart` for computing the partition $\mathcal{P}$ based on the benefit of students from schedules.

Input: requirement $\texttt{req}(s,t)$ for every $s \in \mathbf{S}$ and $t \in \mathbf{T}$, number of timeslots d, number of groups K.
Output: partition $\mathcal{P}$.

$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K\}$
$\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$
$i \in_R [1, 2, \ldots, K]$, $\mathcal{P}_i \leftarrow s$ for every $s \in S$
for $i = 1, \ldots, K$ **do**
    $\mathcal{A}_i = \texttt{Schedule}(P_i, d)$
end for
while convergence is achieved **do**
    for all students $s \in \mathbf{S}$ **do**
        $P_i \leftarrow s$, $i = \text{argmax}_{j=1,\ldots,k}\, \texttt{Benefit}(s, \mathcal{A}_j)$
    end for
    for $i = 1, \ldots, K$ **do**
        $\mathcal{A}_i = \texttt{Schedule}(P_i, d)$
    end for
end while

`RandPart`: Partition S at random.

`K_means`: We represent each student s by the $|T|$-dimensional vector $\big(\texttt{req}(s,t_1), \texttt{req}(s,t_2), \ldots, \texttt{req}(s,t_{|T|})\big)$ containing its requirements for each topic. We assign students to groups based on the `K_means` clustering performed on the space of the requirement vectors using Eucledian distance.

`CohPart_S`: We also investigate a speedup version of `CohPart`. We pick a subset of $n' << n$ students $S' \subset S$ at random. We compute the optimal group schedules $\mathcal{A}'_1, \mathcal{A}'_2, \ldots \mathcal{A}'_K$ for S' using `CohPart` and then assign each student $s \in S$ to the group that maximizes $\texttt{Benefit}(s, \mathcal{A}'_i)$.

5.2 Datasets

BUCS data. This dataset consists of grades of real students who majored in CS at Boston University. The data consists of 398 students and 41 courses. Here the courses correspond to topics and letter grades were converted to the requirement of students. That is, grades A – F were converted to $\texttt{req}(s,t)$ such that A = 5 and F = 50. We assumed the number of requirement to master a course for the smartest student is 5 (base parameter). As the ability drops, number of requirement goes up (step parameter). To compute missing requiements, i.e., fill values for missing (student, course) pairs, we used Graded Response Model (GRM). First, using GRM we obtain the ability and difficulty parameters for all students and all courses. Then for each pair of (student, course) in which student s did not take course c, we used the ability of s and difficulty of c to predict the grade of course c for that student.

BUCSSynth data. In order to see how well our algorithm scales to larger datasets, we generated a synthetic data, based on the obtained parameters from GRM. We call this dataset BUCSSynth. From BUCS dataset, we observed that the ability of students follows a normal distribution with $\mu = 1.13$ and $\sigma = 1.41$. Applying GRM to BUCS, we obtained difficulty parameters for 41 courses. In order to obtain difficulties for 100 courses, we used the following:

1. Choose one of the 41 courses at random.
2. Use density estimation, smoothing and then get the CDF of the difficulties.
3. Randomly sample from the CDF to get the difficulties for a new course.

Using these parameters, we generated grades for 2000 students and 100 courses and we transformed grades to number of requirements similar to what we did for BUCS dataset.

Synthetic data. In ground truth dataset we had generated 10 groups of students, each group containing 40 students. For each group we selected 5 courses and assigned requirement randomly to those 5 courses such that the sum of requirement will be equal to the deadline. Then for the remaining 35 courses, we filled number of requirements with random numbers taken from a normal distribution with $\mu = \frac{deadline}{5}$ and $\sigma = 3$. We refer to this dataset as GroundTruth.

We have also generated the requirements for 400 students and 40 courses using Pareto ($\alpha = 2$), Normal ($\mu = 30$ and $\sigma = 5$) and Uniform (in the range of [5,100]) distributions. We refer to this datastes as `pareto`, `normal` and `uniform`.

5.3 Results

All algorithms are implemented in Python 2.7 and all the experiments are run single threaded on a Macbook Air (OS-X 10.9.4, 4GB RAM). We compare our algorithm with `RandPart`and the `K_means`algorithm, the built in k-means function in Scipy library. Each experiment was repeated 5 times and the average results are reported in this section. For sample size in `CohPart_S` algorithm, we set parameter c (explained earlier) to 4 in all experiments.

5.3.1 Results on Real World Datasets

BUCS. The result on the BUSC data is depicted in Figure 1e where each point shows the benefit of all students when partitioning them into K groups. As we see the `RandPart` has the lowest benefit and our algorithm has the best benefit. As the number of clusters increases (having hence fewer students in each cluster), the benefit also increases, means the schedule for those students is more personalized and closer to their individual schedule. In Figure 1f we show that the greater the deadline is, the closer `K_means` gets to our algorithm. But in real life, we do not have enough time to repeat (or teach) all of the courses (for e.g. for preparation before SAT exam). Figure 1f illustrates the case when deadline is equal to the average sum of need vectors for different students.

BUCSBase. We tried different values for base and step parameters (explained earlier) and the result is depicted in Figure 1g when the base and step are equal to 1. The larger is the value of base and step parameter, the better our algorithm performs.

BUCSSynth dataset. We ran our algorithms on on BUCSSynth dataset to see how well our algorithm scales for large number of students. The result is depicted in Figure 1h.

5.3.2 Results on Synthetic Datasets

The result on synthetic data is illustrated in Figure 1a. As we see `CohPart` and `CohPart_S` both are performing well. For all of the courses the mean requirement is close to 10 with standard deviation 3. We expect that students in the same

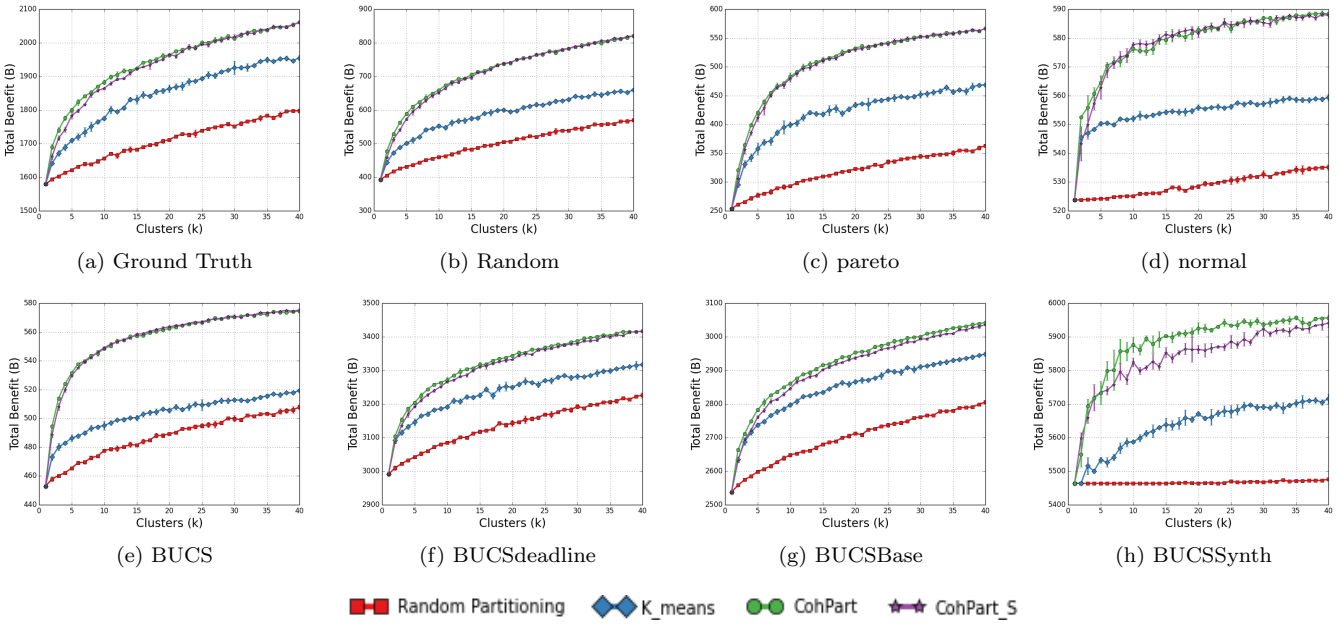


Figure 1: Total benefit achieved by different algorithms as a function of the number of groups of students.

group (when generating the data) should be placed in the same cluster after running our algorithm and the schedule should include the selected courses in each group. Students have different requirement values for the selected courses in each group, but the sum of these selected courses is equal to the deadline and our algorithm realized this structure and only considered these selected courses to obtain the schedule. But `K_means` lacked this ability to find the hidden structure. The next studied datasets were `uniform`, `pareto` and `normal` datasets and the results are depicted in Figure 1b, 1c and 1d respectively. For these datasets also our algorithm outperformed `K_means` and `RandPart` .

6. CONCLUSION

In this paper, we highlighted the importance of team formation and scheduling educational materials for students. We suggested a novel clustering algorithm to form different teams and teach the team members based on their abilities. The results we obtained shows that our proposed solution is effective and suggest that we have to consider personalized teaching for students and form more efficient teams.

7. ACKNOWLEDGMENTS

This work was partially supported by NSF Grants: #1430145, #1414119, #1347522, #1239021, #1012798, #1218437, #1253393, #1320542, #1421759.

8. REFERENCES

[1] Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers & Education*, 68(0):199 – 210, 2013.

[2] R. Agrawal, B. Golshan, and E. Terzi. Grouping students in educational settings. In *ACM SIGKDD*, pages 1017–1026, 2014.

[3] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Power in unity: Forming teams in large-scale community systems. In *ACM International Conference on Information and Knowledge Management*, pages 599–608, 2010.

[4] A. Ashman and R. Gillies. *Cooperative Learning: The Social and Intellectual Outcomes of Learning in Groups*. Taylor & Francis, 2003.

[5] P. Brucker. *Scheduling Algorithms*. Springer-Verlag New York, Inc., 3rd edition, 2001.

[6] R. F. Bruner. Repetition is the first principle of all learning. *Social Science Research Network*, 2001.

[7] D. Esposito. Homogeneous and heterogeneous ability grouping: Principal findings and implications for evaluating and designing more effective educational environments. *Review of Educational Research*, 43(2):163–179, 1973.

[8] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. *J. ACM*, pages 263–280, 2004.

[9] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *ACM SIGKDD*, pages 467–476, 2009.

[10] A. Majumder, S. Datta, and K. Naidu. Capacitated team formation problem on social networks. In *ACM SIGKDD*, pages 1005–1013, 2012.

[11] T. P. Novikoff, J. M. Kleinberg, and S. H. Strogatz. Education of a model student. *Proceedings of the National Academy of Sciences*, 109(6):1868–1873, 2012.

[12] A. Segal, Z. Katzir, K. Gal, G. Shani, and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning. 2014.

[13] R. E. Slavin. Ability Grouping and Student Achievement in Elementary Schools: A Best-Evidence Synthesis. *Review of Educational Research*, 57(3):293–336, 1987.

An approach of collaboration analytics in MOOCs using social network analysis and influence diagrams

Antonio R. Anaya
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913986550
arodriguez@dia.uned.es

Jesús G. Boticario
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913989387
jgb@dia.uned.es

Emilio Letón
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913989473
emilio.leton@dia.uned.es

Félix Hernández-del-Olmo
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913988345
felixh@dia.uned.es

ABSTRACT
MOOCs pedagogical strategies assume that students construct their own knowledge and collaborate with their mates. Large-scale learners' interaction figures hinder both proper interpretation of learners' needs and prompt remediation actions. To this we describe a preliminary study of a two-step collaboration analysis, which consists of inferring domain-independent indicators on students' relationships obtained from social network analysis and using an influence diagram to warn teachers on students' problematic circumstances to facilitate prompt remediation actions.

Keywords
Collaboration analytics, SNA, influence diagram, collaborative learning

1. INTRODUCTION
Massive open online courses (MOOCs) are stood out as a new pedagogical methodology since they aimed at large-scale participation and open access via the web [1]. In this situation the teacher loses control over the learning process and students should construct their own learning. The students can use the MOOC's communication means to collaborate with their learning mates [2]. In this respect, although the students are to be provided with the tools and services to collaborate, this thus not suffice and frequent and regular analyses of the team process are needed to know whether the collaboration takes place [3]. Moreover, the special large-scale nature of MOOCs hampers teachers when coming to analyze students' communication acts, which drive the collaboration process.

Some researchers have proposed a well-known analysis method, social network analysis (SNA) to minimize the problems commented above [4, 2]. However, in this collaborative learning context some variables, such as emotion and empathy, are out of control [3]. Under these circumstances, analyzing the collaboration process requires to deal with uncertainty [5], which can be tackle with Influence diagrams (ID) [6].

In our research we propose an approach to automatically warn (or recommend [7] teachers on students' problematic collaboration circumstances so that they can readily provide corrective actions when required. Thus, the objectives of the application are: 1) to analyze the collaboration with a transferable analysis method that provides domain-independent collaborative indicators; 2) to minimize the human intervention.

The rest of the paper is organized as follows. First we describe in Section 2 related research, to both SNA in MOOCs and ID in the educational context. In Section 3 we frame the research and educational context in which this work is being applied and an in-depth description of the proposed methodology. We then comment on our preliminary study in Section 4 and finally briefly provide the main conclusions and further planned research in Section 5.

2. Related research
MOOCs offer more leeway to students and thereof features new challenges [8]. In this more crowded and less constrained learning environment it is advisable to use any available technology to analyze the learning process involved. Here technologies such as SNA are starting to be applied with relative success [2].

SNA has been used to identify students who are actively participating in course discussions and thus are potentially at a risk of dropping out [2]. [4] examined and detected, using SNA, communities of users within a large course so that they can be provided with a personalized and social-oriented recommender system. [9] presented an example of a Social Learning Analytics Tool to visualize real-time discussion activities in a MOOC environment.

SNA has been widely applied to study the social aspect of students learning [10]. This way [11] analyzed networks in order to identify the people from whom an individual learns. Here [12] proposed a methodology to analyze students' interactions in a collaborative learning environment, which consists of using SNA to get meaningful statistical indicators, such as the student reputation. [13] emphasized the use of SNA techniques to discover relevant structures in social networks so that the instructors were able to better assess participation.

As the aforementioned approaches we aimed at improving collaborative settings though SNA outcomes in terms of a technology that has proved its usefulness in tackling problems under uncertainty. Moreover, the educational context has been a traditional suitable field where Bayesian networks (BN) have been applied to deal with the inherent uncertainty involved [14]. [15] proposed a course diagram method, based on an ID framework, which can be used by an instructor to design a course structure. The diagram organizes the instructional material and the tests.

3. Towards collaboration analytics in MOOCs
In our education context we proposed to combine two different technologies to analyze the collaboration. Firstly, the SNA obtains indicators from students' interactions, which reflects how students connect with their mates. Secondly, an influence diagram

structures students' indicators as a network, which supports a decision on students' problematic circumstances once an expert, who can be the tutor, tunes the probabilities of the network. The software used for SNA was Gephi[1] and for ID was OpenMarkov[2].

3.1 Social network analysis

To date the most common communication service in MOOCs is forums. SNA has been applied to forums in order to infer the social relationships among users [16]. Here SNA metrics support the inference of social relationship indicators.

Figure 1 shows the SNA diagram resulting from the data of the on-line course that we have used in the preliminary study.

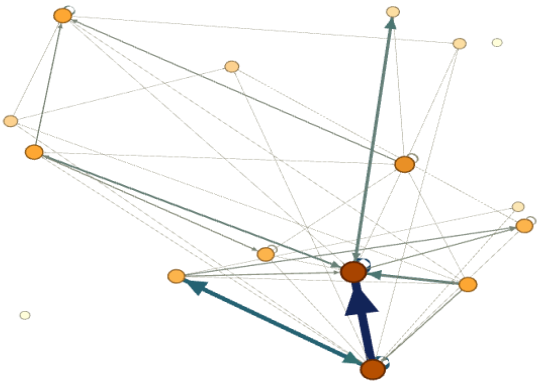


Figure 1. SNA in the preliminary study.

In Figure 1 nodes are students who participated in an online course (see Preliminary study section and communications among students were analyzed through SNA. Within this figure the metric Degree of the nodes is represented as follows: red and big node means high degree, and yellow and small node means low degree. The color and size of the ties mean the weight of the relationship (i.e., number of messages from origin node to destiny node).

We propose the following centrality metrics of the nodes as indicators of the collaboration process:

- **Degree** is the number of ties of one node.
- **In-degree** is the number of ties whose destiny is the node. This indicator is a measure of the node popularity.
- **Out-degree** is the number of ties whose origin is the node. This indicator is a measure of the node sociability.
- **Closeness centrality** is the degree to which an individual is near all other individual in the network. This reflects the ability to access to information by the network members.
- **Betweenness centrality** a measure that quantifies the frequency or number of times that a node acting as a bridge along the shortest path between two other nodes.
- **Eigenvector centrality** is the measure of the importance of a node in the network. Intuitively, the nodes that have a high value of this measure of centrality are connected to many nodes, which are connected also in this sense; therefore, are good candidates to disseminate information.

We use these indictors, because they are well-known in the state-of-the-art research focused on analyzing the position of the students in the network using SNA [16]. These indicators constitute a standard way to measuring network and node features and they can be used in several different context.

3.2 Influence diagrams

IDs provide us with a framework for representing and solving decision problems under uncertainty. As our objective is to maintain a domain independent and general approach of inference IDs include features that are advisable in learning environment as MOOCs, where the collaborative learning is encouraged. The collaboration settings constitute a framework where not all variable are known in advance. In addition, a MOOC is an educational environment where teachers cannot afford the continuous tracking and analysis phases of learners' interactions, which in this case are massive. An ID could help teachers to identify and carried out correction decisions adapted to each student.

We propose an ID where the indicators obtained from the SNA, the centrality metrics commented above, are structured. The network layout of the proposed ID is showed in Figure 2.

In Figure 2 the yellow and round nodes are the variables in the problem. Assessment is the root and hidden variable, which is unknown in future test. The node "Assessment" represents the teacher's assessment of students' collaboration. The ID needs a training dataset with known values of the node "Assessment" to tune the networks probabilities. The other yellow and round nodes are the SNA indicators. The squared node "D" represents the decision, in this case, yes or not. The decision "yes" means a detection of problematic circumstances and the ID supports teacher with a suggestion so that the teacher makes a corrective actions. The node "U" maximizes the decision utility. Notice that the values of the nodes have to be discretized. In order to do the discretization we divided interval values into three groups with equal width. We propose three values: high, medium and low, because these values are easy to understand.

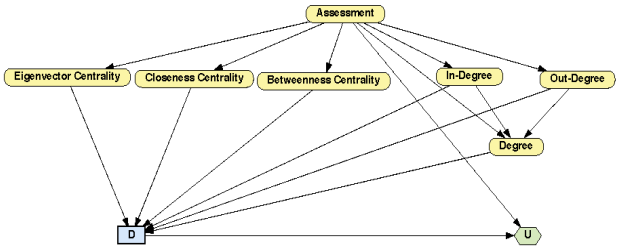


Figure 2. Network of the influence diagram

4. Preliminary study

In the preliminary study we have used data from an on-line course to fine-tune the ID's network. The experience was done with students of the subject "Complexity and Computability" in the forth course of the degree of Computer Systems Engineering at UNED (Spanish National University for Distance Education). In this subject we have mimicked the characteristics of MOOCs, with particular emphasis on the participation on the forum. For that reason we have undertaken a continuous assessment process on the Learning Management System (LMS) forum's interactions

[1] http://gephi.github.io

[2] http://www.openmarkov.org

in order to detect the student level of participation and the recording of a special type of video podcast [17].

We have tested our approach in an online course, which let us make a preliminary proof of concept on the main issues involved, namely tracking and assessing students (16 students). This curse has been designed following the large-scale MOOC's course settings, meaning that it consists of the same video lectures, individual tasks and a communication services that will be ultimately provided [17].

Node Potential: Degree

Relation Type: Table

Assessment	low	low	low	low	low
Out-Degree	low	low	low	middle	middle
In-Degree	low	middle	high	low	middle
high	0	0.333333	0.333333	0	0.333333
middle	0	0.333333	0.333333	0	0.333333
low	1	0.333333	0.333333	1	0.333333

Figure 3: An example of node "Degree" probabilities.

In the fine-tuning process experts can insert knowledge into the ID's network, that is, in the automatic inferring process. Firstly the students should be assessed according to their interactions. It is fairly common that experts decide which students' features, that their interactions have revealed, are the most relevant to be assessed. This knowledge is showed when the assessments are compared with the analysis of students´ interactions, which is independent of expert's assessments. We made the SNA of the students´ interactions and independently an expert assessed the students.

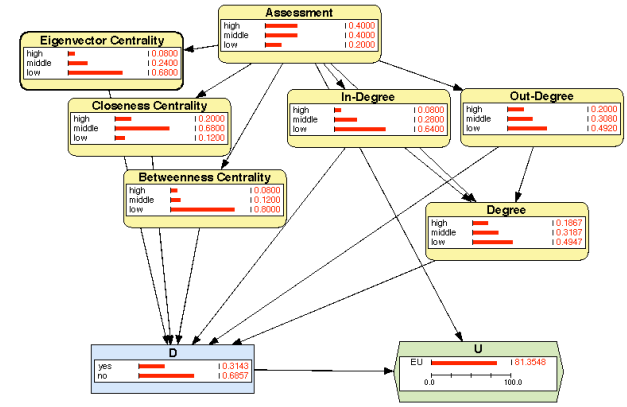


Figure 4: A general perspective of the ID's network results.

Once the students were assessed and we obtained the SNA centrality attributes of each student, we then discretized the data. Then, we were able to measure the probabilities for each case. An example is show in Figure 3. According to the Figure 2, the node "Degree" has three fathers, the nodes "In-Degree", "Out-Degree" and "Assessment". For each possibility of "Degree" value (low, medium or high) we measured the probability according to the values of the father nodes. Figure 3 shows some cases. For instance, when the father node have "low" value, the node "Degree" have "low" value. Because node "Degree" has three father nodes, there are 27 possible cases (the Cartesian product of three variables with three possible values). We made the fine-tuning process with 16 students, thus, we did not have enough data to fine-tune the network completely. We could solve this lack with data from the next experience.

After the probabilities were established for each possible case of each node, the ID was able to infer a decision and the decision utility for each case. Figure 4 shows the general perspective ID's results. It can be seen that the ID advises to recommend only in around one third of cases (In node "D", "yes" is 0.3143 and "no" 0.6857).

We can observe what happens when the ID advises to recommend, i.e., identifies a possible collaboration problematic circumstance. Figure 5 shows the case when the ID advises to recommend. When the ID advises to recommend, the student has low value in the nodes "Degree" and "In-Degree". This informs us that when a recommendation is advisable, the student is not active and her/his classmates ignore her/him. Thus, the ID has identified a problematic collaboration scene, which can be happened over the course. With this information the teacher could make a corrective activity to improve the collaboration process.

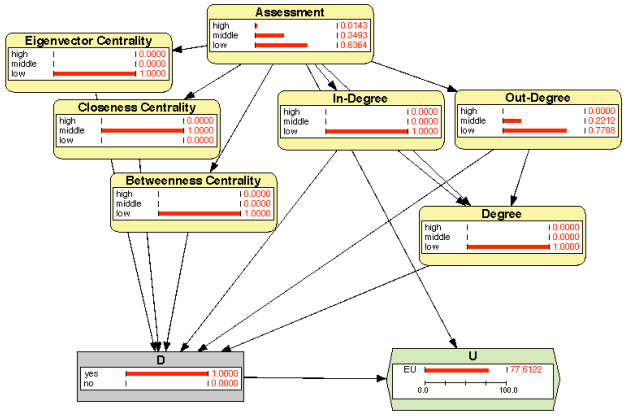


Figure 5. Example: ID advises to recommend.

In addition to the previous analysis, it is possible to calculate the optimal policy of the ID (see Figure 6). Thus, the optimal policy informs about the decision (yes or not) for each combination of nodes values.

Optimal policy: D

Relation Type: Table Reorder variables

Degree	low	low	low	low	low	low	low	low
Eigenvector Centrality	low	low	low	low	low	low	low	low
Betweenness Centrality	low	low	low	low	low	low	low	low
Out-Degree	low	low	low	low	low	low	low	low
Closeness Centrality	low	low	low	middle	middle	middle	high	high
In-Degree	low	middle	high	low	middle	high	low	middle
yes	0	0.5	0.5	1	0	0	0	0
no	1	0.5	0.5	0	1	1	1	1

Figure 6. Optimal policy: all possible decisions of ID.

Figure 6 shows an example of the decisions, yes or not, according to the values (high, middle or low) of the centrality attributes obtained from the SNA. In the preliminary study we had 16 assessed students. The possible cases that the ID can consider mathematically are the Cartesian product of network nodes and the student indicators (a total of 729 cases). Thus, not all the possible cases of the nodes values combination have to be considered by the ID. However, the results (see Figure 6) show that the ID is capable to support with different decisions according to the students SNA centrality attributes values. However, more

interaction data are needed to continue with the ID tuning process. When tuning process is finished, a new student's attributes values from the SNA feed the ID that, in turn, can offer accordingly a new decision (i.e., "yes", suggestion of a corrective action due the possible student's problematic circumstance in the collaboration).

The approach labels students with "yes" (the student needs a recommendation) or "not" (the student does not) and this way guides teachers to identify the student's collaboration problem. Based on this the teacher can create the appropriate recommendation to the student.

5. Conclusions and future work

To facilitate collaborative learning management within MOOCs in this paper we propose a domain independent and transferable approach, which is based on two different technologies: 1) Inferring domain-independent indicators on students' relationships obtained from social network analysis (SNA) in their interactions; 2) From these indicators an ID is used to warn teachers on students' problematic circumstances so they can provide them with prompt remediation actions. Here teachers cannot afford the continuous tracking and analysis phases of learners' interactions, which in this case are massive.

The preliminary results described in this paper confirm that the approach can identify problematic collaboration scenes, although it should be further investigated. Thus, data from more students will be considered, which will be used to tune the ID's network probabilities. Thanks to the approach, the tuning process can be made while the students are participating in the MOOC. Moreover, the final suggestion that is offered to the teacher can also be improved. The suggestion should be easily understandable by any non-expert user so that the analysis process involved won't prevent them from its usage.

The research described in this paper will be further applied within the MAMIPEC project, which aimed to infer and provide affective personalized support to learners in educational contexts [18].

6. ACKNOWLEDGMENTS

Authors would like to thank the MAMIPEC project (TIN2011-29221-C03-01), which has been funded by the Spanish Ministry of Economy and Competence.

7. REFERENCES

[1] Masters, K. (2011). A brief guide to understanding MOOCs. The Internet Journal of Medical Education, 1(2).

[2] Sinha, T. (2014, February). Together we stand, Together we fall, Together we win: Dynamic team formation in massive open online courses. In Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the (pp. 107-112). IEEE.

[3] Johnson, D., Johnson, R. (2004). Cooperation and the use of technology. In: Handbook of research on educational communications and technology. Taylor and Francis Group, pp. 401–424.

[4] Zhuhadar, L. and Butterfield, J. (2014). Analyzing Students Logs in Open Online Courses Using SNA Techniques, Twentieth Americas Conference on Information Systems, Savannah, 2014.

[5] Anaya, A. R., Luque, M., García-Saiz, T. (2013). Recommender system in collaborative learning environment using an influence diagram. Expert Systems with Applications 40, 7193–7202.

[6] Howard, R. A., Matheson, J. E. (1984). Influence diagrams. In: Howard, R. A., Matheson, J. E. (Eds.), Readings on the Principles and Applications of Decision Analysis. Strategic Decisions Group, Menlo Park, CA, pp. 719–762.

[7] Hernández del Olmo, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. Expert Systems with Applications, 35(3), 790-804.

[8] Duque, R., Bravo, C., Ortega, M. (2013). An ontological approach to automating collaboration and interaction analysis in groupware systems. Knowledge-Based Systems 37, 211 – 229.

[9] Schreurs , B., Teplovs, C., Voogd, S. (2014). Social Learning Analytics applied in a MOOC-environment, eLearning Papers, 36, January 2014

[10] Shum, S. B. and Ferguson, R. (2012). Social Learning Analytics. Educational Technology & Society, 15, 3—26

[11] Haythornthwaite, C., & De Laat, M. (2010). Social networks and learning networks: using social network perspectives to understand social learning. Paper presented at the 7th International Conference on Networked Learning, Aalborg, Denmark.

[12] Bratitsis, T., Dimitracopoulou, A., Martínez-Monés, A., Marcos-García, J., Dimitriadis, Y. (2008). Supporting members of a learning community using interaction analysis tools: the example of the Kaleidoscope noe scientific network. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT 2008. pp. 809–813.

[13] Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). Analyzing participation of students in online courses using social network analysis techniques. In Proceedings of educational data mining.

[14] Millán, E., Loboda, T., de-la Cruz, J. P. (2010). Bayesian networks for student model engineering. Computers & Education 55, 1663–1683.

[15] Chang, F.-I. (2003). Quantitative analysis on distance learning courseware. Multimedia Tools and Applications 20, 51–65.

[16] Dawson, S. (2008). A Study of the Relationship between Student Social Networks and Sense of Community, Educational Technology & Society, 11, 224—238

[17] León, E., Molanes-López, E.M. (2014). Two New Concepts in Video Podcasts: Minimalist Slides and Modular Teaching Mini-videos. 6th International Conference on Computer Supported Education.

[18] Santos, O.C., Salmeron-Majadas, S., Boticario, J.G. (2013). Emotions Detection from Math Exercises by Combining Several Data Sources. AIED 2013, Lecture Notes in Computer Science LNCS/LNAI 7926, 742–745.

On Convergence of Cognitive and Noncognitive Behavior in Collaborative Activity

Luna Bazaldua, Diego A.
Teachers College, Columbia University
dal2159@tc.columbia.edu

Khan, Saad
Educational Testing Services
skhan002@ets.org

von Davier, Alina A.
Educational Testing Services
avondavier@ets.org

Hao, Jiangang
Educational Testing Services
jhao@ets.org

Liu, Lei
Educational Testing Services
lliu@ets.org

Wang, Zuowei
University of Michigan
zwwang@umich.edu

ABSTRACT
We present results from a pilot study to investigate the evidence for convergence and synchrony in cognitive and noncognitive behavior of dyads engaged in a collaborative activity. Our approach utilizes multimodal data including video and participant action log files retrieved from the collaborative activity, an online educational simulation on science topics. The log files captured cognitive behavior including frequency and content of chat messages between dyads, as well system help requests. The video data recorded participant nonverbal behavior that was processed on a frame-by-frame basis using automated facial expression classifiers and coded by trained human raters on high-level noncognitive behaviors including: affect display gestures, engagement, anxiety and curiosity. The data were analyzed at individual and dyad levels and results using hierarchical clustering analysis demonstrate evidence of cognitive and noncognitive behavioral convergence among dyads.

Keywords
Collaborative Assessment, Human-Computer Interaction, Multimodal Data, Noncognitive states, Cluster Analysis

1. INTRODUCTION
Behavioral convergence refers to the unintentional imitation process of gestures, facial expressions, behaviors, moods, postures, or verbal patterns of coparticipants on a range of different time-scales [4, 12]. In literature it has been referred to by a variety of terms e.g., behavioral matching, mimicry, interpersonal coordination, entrainment, interactional synchrony and the Chameleon effect [4, 12, 17, 19]. While previous studies have explored its impact on interpersonal skills, coordinated activity, negotiations, and how individuals influence the behaviors of others [2, 4, 21], little research has focused on finding evidence for behavioral convergence in collaborative activity [24].

Collaboration is a complex activity that constitutes an interplay between *cognitive processes* such as knowledge acquisition, content understanding, action planning, and execution [7, 8, 10, 18, 26] and *noncognitive processes* such as social regulation, adaptability, engagement and social affect, such as boredom, confusion, and frustration [1, 3, 6]. Collaborative activity may take place in face-to-face interactions or through the medium of online distance learning technologies and collaboration platforms [20]. In either context collaboration is more effective when participants are engaged in the task and exhibit behaviors that facilitate interaction [25].

Our hypothesis is that behavioral convergence occurs during collaborative activity and it manifests in both cognitive and noncognitive processes. Based on this premise, we expect that people will tend to synchronize their behaviors (consciously or nonconsciously) while they are engaged in a collaborative activity. To test our hypothesis, a pilot study was conducted involving 12 unique dyads collaborating in an online game-like science assessment: ETS' online collaborative research environment ─the Tetralogue [15, 27]. Multimodal data including video and activity log files of each participating dyad were captured. The log files contain cognitive behavior including frequency and content of chat messages between dyads, as well as system help request (i.e., the participant requests to view educational videos on the subject matter to better answer assessment questions). The video data, on the other hand, recorded participant nonverbal behavior which was analyzed on a frame-by-frame basis using automated facial expression classifiers and annotated by trained human raters on high-level noncognitive behaviors including: affect display gestures, engagement, anxiety, and curiosity. Along with recent studies [17, 20, 24], in this paper we describe one of the first attempts to capture and analyze multimodal data in the context of studying behavioral convergence in collaborative activities.

2. Methodology
2.1 Collaborative Activity Platform
As mentioned earlier, our study used an online collaboration assessment platform: ETS' online collaborative research environment –the Tetralogue. This platform includes a set of multiple-choice items on general science topics, a simulation based assessment, a personality test, and a set of background questionnaires. The simulation task is on geology topics. The simulation-based task was developed as a task for individual test takers who will interact with two avatars and as a collaborative task that requires the collaboration among two human participants and two avatars in order to solve geology problems.

The participants, who may be in different locations, interact through an online chat box and system help requests (selecting to

view educational videos on the subject matter). The main avatar, Dr. Garcia, introduces information on volcanoes, facilitates the simulation, and requires the participants to answer a set of individual and group questions and tasks. A second avatar, Art, takes the role of another student, in order to contrast his information with that produced by the dyad.

The system logs activity data of the participants in structured XML files, which capture participant actions including: identification of the user who performed each action, the number of chat messages, the content of those chat messages, the number of times the participants request additional information on subject matter from the system, the answer selected for each individual and group question, and the time at which each action occurred.

While the dyads interacted with the task, we captured the video of each individual participant. The video data were used for both annotating noncognitive behavior of the participants and automated facial expression analysis (see section 2.3 for further details). It should be noted that the only form of direct communication between the dyads was through the Tetralogue text-based chat interface and the dyads were not able to see or hear each other. Figure 1 illustrates the collaborative activity and data capture while participants interact in the system.

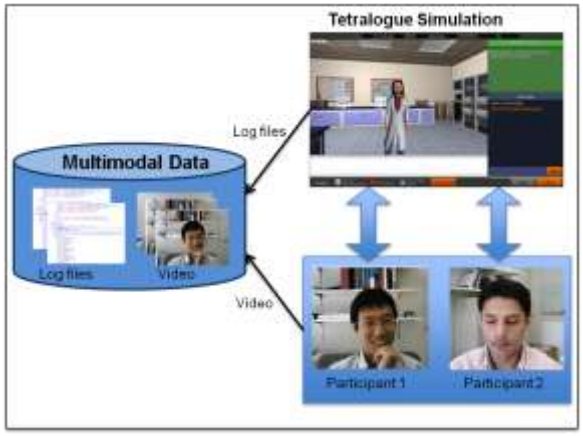


Figure 1. Multimodal data capture including video and action log files while participants engage in collaborative activity on the Tetralogue platform.

2.2 Study Participants and Data Collection

Twenty-four subjects participated in this study and were paired in dyads using random selection. Information about the study was provided to each participant individually and consent forms were obtained from them.

The length of the experiment sessions varied from 15 minutes to 48 minutes, with an average length of 25 minutes. Although there were time variations among sessions, all dyads reviewed the same material and completed the same tasks in Tetralogue. This resulted in approximately 600 minutes of video data and associated participant action log file data. The data stored in the log files were parsed using the 'XML' package [13]. The features extracted from the log files were: number of chat messages sent to the partner and number system helps (viewing educational videos on the subject matter) requested at each stage of the simulation, answer to each individual question, and answer to each group question.

Our focus on "number of messages" and "number of help requests" was driven by former research in the field that associates both features with the performance in learning-oriented tasks, cognitive states, and collaborative interactions [6, 17]. However, more features associated with cognitive activity can be mined from the log files, such as the time length between actions or the content of the chat messages and will be addressed in future studies.

2.3 Video Data Processing and Coding

Facial expression analysis of the video data was performed using the FACET SDK, a commercial version of the Computer Expression Recognition Toolbox [14]. This tool recognizes fine-grained facial features, or facial action units (AUs), described in the Facial Action Coding System [9]. FACET detects human faces in a video frame, locates and tracks facial features, and uses support vector machine based classifiers to output frame-by-frame detection probabilities of a set of facial expressions: anger, joy, contempt and surprise.

In addition, seven trained coders reviewed and coded the videos using the Anvil software [11]. The video data of each participant were assigned to two raters for annotation; however, in three cases there were three raters coding the same video file, and in two cases only a single rater was available for annotation. The raters followed the same coding scheme during the annotation process, which included the next categories: having their hand on their face, expressing engagement, anxiety, or curiosity. As an outcome of the annotation process, the Anvil software produced XML files that were parsed using the 'XML' package [13] in R [22].

Engagement, anxiety, and curiosity were included in the annotation scheme because of the incidence and relevance of these three noncognitive states in simulation games and online learning systems [1, 5]. The coding also included "hand touching face", an affect display gesture that has been linked to affective and cognitive states such as boredom, engagement, and thinking [16].

3. Results

3.1 Behavioral Convergence within Dyads

In order to study evidence of behavioral convergence, features from log files and video data of each of the 24 study participants were represented as a multidimensional behavioral feature vector composed of both the cognitive behaviors: *number_of_messages*, *number_of_help_requests* and the noncognitive behaviors (i.e. fraction of the time each participant exhibited the behavior): *engagment*, *hand_on_face*, *anxiety*, *curiosity*, *anger*, *joy*, *contempt and surprise*.

An agglomerative hierarchical cluster analysis using an average linkage function was performed on an Euclidean distance matrix (i.e., a similarity matrix) computed from the multidimensional behavioral feature data of the study participants. Our hypothesis is that behavioral convergence will manifest in the cognitive and noncognitive features such that members of the same dyad will tend to group together from the beginning of the clustering process (i.e., they will be closer to each other in the feature space than to others).

Figure 2 depicts the dendrogram plot produced from the cluster analysis. In the plot, members of the same dyad are depicted by consecutive numbers and identical color; for instance, the first

dyad includes coparticipants d1.1 and d1.2 colored in red, the second dyad consists of coparticipants d2.1 and d2.2 colored in blue, and so on. The plot shows that participants in 7 of the 12 dyads grouped together in the clustering process (i.e. they were closest to each other in the multidimensional feature space), indicating a high degree of behavioral convergence. Still, some participants (e.g., d10.1 and d4.2) showed a distinctive pattern of values in the variables used to calculate the distances, which prevented them to be grouped with their respective peers.

In addition, we analyzed the similarity matrix of behavioral feature distances for participants within and outside dyads. Behavioral convergence would imply that for dyad members the average distances in feature space is smaller in a statistically significant manner than those of non-dyad members. To study the relative impact of cognitive and noncognitive features we computed two additional similarity matrices: one using exclusively the cognitive features from log files (number of chats messages and number of system help requests) and the other using exclusively noncognitive features produced from the video data (the four facial expression detectors, and the four features from the coding scheme). All features were normalized to present equivalent scaled values between zero and one.

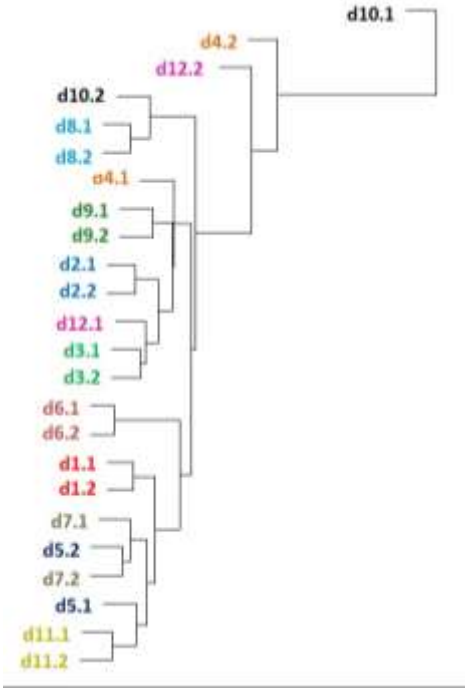


Figure 2. Agglomerative Cluster Dendrogram.

Table 1 shows the mean and standard deviations of feature similarity distances of participants when compared with their dyad partners and others. The results consistently show smaller average distances for the dyads (i.e., members within dyads displayed behavior that was more similar to each other than others), supporting the convergence premise. Additional analysis to test the significance of these differences using the Student's *t*-test demonstrated that using both cognitive and noncognitive features the average distances are statistically significant (t-value = 2.33, df = 11.7, *p*-value < 0.02).

A final analysis was computed on the correlation of the total group scores in the task with the distances of participants with their respective dyad partners and with other users except for their teammate. The group score showed a mild correlation with the distance between dyad members of -0.19 ($s.e._r$ = 0.21). Note that the negative correlation is a consequence of using similarity distances (smaller distance values indicate more convergence) and the group score values (higher values indicate a better performance on the task). Nevertheless, as will be underscored in Section 4, the small sample size in the study produced large standard errors for this correlation estimate and do not imply statistically significant patterns.

Table 1. Average and standard deviation of behavioral feature distances within and outside dyads

Features		Mean	S.D.
Cognitive and noncognitive	**Dyad**	0.57	0.22
	Others	0.73	0.24
Cognitive only	**Dyad**	0.36	0.21
	Others	0.57	0.20
Noncognitive only	**Dyad**	0.41	0.17
	Others	0.41	0.22

4. Discussion and Conclusions

Seminal work from Roschelle [23] in his seminal work made the argument that the crux of learning by collaboration is convergence and showed empirical evidence of the convergence occurring at the linguistic level. Our study provides further empirical evidence of behavioral convergence gleaned from multimodal data. As pointed out in [8], cognitive and noncognitive processes occur simultaneously throughout the collaborative task, and both dimensions cannot be separated in practice. The results from cluster analysis in our experimental study support this idea and the pattern of agglomeration of the participants could be interpreted as evidence of convergence of cognitive and noncognitive states when people interact in a collaborative task.

As reported in table 1, the degree of behavioral similarity within dyads tended to be significantly higher than the similarity between non-dyad members, which is good evidence for behavioral convergence in collaborative interactions [4, 12]. In addition, we observed a mild correlation (of approximately 0.2) between the measure of convergence (i.e., the level of similarity between dyads) and the dyad task scores. This might be interpreted as a scaffolding effect that convergence during interaction can have in group performance outcomes. Similar results were reported in [24], underscoring that specific types of convergence have a positive effect in learning and collaboration.

Further research using these data will address topics such as the synchrony of behavior and noncognitive states between members within dyads, machine learning and classification analyses to detect and predict specific cognitive and noncognitive states from facial action units, and more detailed analysis on the impact of cognitive and noncognitive states on the individual-level and group-level assessment outcomes.

There are certain limitations of this study that should be pointed out. First, the current sample size is small ─24 participants─ despite the rich amount of information gathered from each participant. Second, the current collaboration platform neither allows participants to view each other nor uses face-to-face audio-visual interfaces to communicate. This limits how participants are able to mirror each other's behavior and may also explain why we observed weaker convergence in noncognitive features. Third, the

study has utilized a very limited set of behaviors both cognitive and noncognitive. We aim to extend our behavior feature set and sources of data (e.g., audio data) in future studies as well as utilize the content of participant chat messages to glean features like shared vocabulary, turn-taking etc.

5. REFERENCES

[1] R. Baker, S.K. D'Mello, M.M.T. Rodrigo, & A.C. Graesser. 2010. Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies*, 68, 4, 223-241.

[2] S. Barsade. 2002. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, 47, 644–675.

[3] M. Ben Ammar, M. Neji, A. M. Alimi, & G. Gouardères. 2010. The affective tutoring system. *Expert Systems with Applications*, 37, 4, 3013-3023.

[4] S. Bilakhia, S. Petridis, & M. Pantic. 2013. Audiovisual detection of behavioural mimicry. In *Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction*. 123-128.

[5] R.A. Calvo, & S. D'Mello. 2010. Affect detection: An interdisciplinary review of models, methods, and their applications. *Transactions on Affective Computing*, 1, 1, 18-37.

[6] A.C. Graesser, S.K. D'Mello, S.D. Craig, A. Witherspoon, J. Sullins, B. McDaniel, & B. Gholson. 2008. The relationship between affective states and dialog patterns during interactions with AutoTutor. *Journal of Interactive Learning Research*, 19, 2, 293-312.

[7] S. Greiff. 2012. From interactive to collaborative problem solving: Current issues in the Programme for International Student Assessment. *Review of Psychology*, 19, 2, 111–121.

[8] J. Hao, L. Liu, A.A. von Davier, & P. Kyllonen. 2015. Assessing collaborative problem solving with simulation based task. Paper to be presented at the *11th international conference on computer supported collaborative learning*.

[9] J.C. Hager, P. Ekman, & W.V. Friesen. 2002. *Facial action coding system*. A Human Face, Salt Lake City, UT.

[10] A. Hron, & H.F. Friedrich. 2003. A review of web-based collaborative learning: factors beyond technology. *Journal of Computer Assisted Learning*, 19, 1, 70-79.

[11] M. Kipp. 2012. Multimedia Annotation, Querying and Analysis in ANVIL. In *Multimedia Information Extraction*. M. Maybury, Ed., Wiley-IEEE Computer Society Press, 351-367.

[12] J.L. Lakin, V.E. Jefferis, C.M. Cheng, & T.L. Chartrand. 2003. The chameleon effect as social glue: Evidence for the evolutionary significance of nonconscious mimicry. *Journal of nonverbal behavior*, 27, 3, 145-162.

[13] D. T. Lang. 2013. *XML: Tools for parsing and generating XML within R and S-Plus*. R package version 3.98-1.1. http://CRAN.R-project.org/package=XML.

[14] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, & M. Bartlett. 2011. The computer expression recognition toolbox (CERT). In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*. 298-305.

[15] L. Liu, J. Hao, A.A. von Davier, P. Kyllonen, & D. Zapata-Rivera. In press. A tough nut to crack: Measuring collaborative problem solving. In *Handbook of Research on Computational Tools for Real-World Skill Development*. Y. Rosen, S. Ferrara, & M. Mosharraf, Eds., IGI-Global, Hershey.

[16] M. Mahmoud, & P. Robinson. 2011. Interpreting Hand Over Face Gestures. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 248-255.

[17] R. Martinez, J. Kay, J. Wallace, & K. Yacef. 2011. Modelling and identifying collaborative situations in a collocated multi-display groupware setting. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, 196-204.

[18] H.F. O'Neil, S.H. Chuang, & G.K.W.K. Chung. 2003. Issues in the computer-based assessment of collaborative problem solving. *Assessment in Education*, 10, 361-373.

[19] J.S. Pardo. 2006. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119, 4, 2382-2393.

[20] D. N. Prata, R. Baker, E. Costa, C. P. Rosé, Y. Cui, & A.M.J.B. de Carvalho. 2009. Detecting and Understanding the Impact of Cognitive and Interpersonal Conflict in Computer Supported Collaborative Learning Environments. In *Proceedings of the 2nd International Conference on Educational Data Mining*. 131-140.

[21] A. Pentland. 2008. *Honest Signals: How they shape our world*. MIT Press, Cambridge, MA.

[22] R Core Team. 2012. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[23] J. Roschelle. 1992. Learning by collaborating: Convergent conceptual change. *The journal of the learning sciences*, 2, 3, 235-276.

[24] B. Schneider. 2014. Toward Collaboration Sensing: Multimodal Detection of the Chameleon Effect in Collaborative Learning Settings. In *Proceedings of the 7th International Conference on Educational Data Mining*, 435-437.

[25] A.A. Tawfik, L. Sanchez, & D. Saparova. 2014. The Effects of Case Libraries in Supporting Collaborative Problem-Solving in an Online Learning. *Environment. Technology, Knowledge and Learning*, 19, 3, 337-358.

[26] A.A. von Davier, & P.F. Halpin. 2013. *Collaborative Problem Solving and the Assessment of Cognitive Skills: Psychometric Considerations*. ETS Research Report No.RR-13-41. Educational Testing Service, Princeton, NJ.

[27] D. Zapata-Rivera, T. Jackson, L. Liu, M. Bertling, M. Vezzu, & I.R. Katz. 2014. Assessing science inquiry skills using trialogues. In *Intelligent Tutoring Systems*. S. Trausan-Matu, K. Boyer, M. Crosby, & K. Panourgia, Eds., Springer International Publishing, 625-626.

The Impact of Small Learning Group Composition on Student Engagement and Success in a MOOC

Zhilin Zheng*
Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
zhilin.zheng@hu-berlin.de

Tim Vogelsang*
iversity GmbH
Bernau bei Berlin, Germany
t.vogelsang@iversity.org

Niels Pinkwart
Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
niels.pinkwart@hu-berlin.de

ABSTRACT
A commonly known and widely studied problem of massive open online courses (MOOCs) is the high drop-out rate of students. In this paper we propose and analyze the composition of small learning groups as a solution to this problem. In an experiment, we composed such small learning groups in a MOOC context using two methods: Random grouping and grouping by an algorithm that considers selected student criteria. Further, a flipped classroom course was conducted on-campus with a local student group using the MOOC. We compared all three approaches to a control condition using two measures: Drop-out rate and learning performance. The empirical results give an indication, yet no hard evidence, that small groups might reduce student drop-out rates.

Keywords
MOOC; Group Composition; Learning Analytics; Drop-out Rate.

1. INTRODUCTION
MOOC providers, such as Coursera, EdX and iversity, reach course enrolments of up to tens of thousands of students using scalable techniques like lecture videos and quizzes [7]. This massive scale reduces the opportunities for interaction with course instructors. Completion rate, a commonly used (yet debatable) measure of student success, is reported to be less than 13 percent in most MOOCs [3], which has recently attracted extensive studies in order to discover reasons behind this problem [5; 8; 11]. Social connections and collaboration between MOOC students also fall far below expectations. Only 5-10 percent actively participate in course forums [9]. At this point, group formation might help by leading to the creation of informal social ties [4] as well as improving social skills [10].

The composition of small learning groups has already been tested in online learning contexts and local meeting scenarios (i.e. face-to-face groups). In general, self-selected, random and algorithm-based group composition are commonly applied. Algorithm composed groups typically bring together students with either heterogeneous or homogeneous criteria (e.g. based on learning style, personality and demographic information) using technologies such as GT [1] or Swarm Intelligence [2]. Unlike the case with randomly composed or self-selected groups, students' information must be preliminarily collected and then provided to the composition algorithm.

In order to investigate the impact of small learning groups on drop-out rate and learning performance, we conducted a grouping experiment on the iversity.org platform. Specifically, we tested three grouping approaches, all in the same MOOC: 1) automated group composition using an adapted k-means clustering algorithm, accounting for both homogeneous and heterogeneous student criteria; 2) random group composition; and 3) an on-campus flipped classroom approach. This paper describes the results in the three conditions concerning drop-out rates, learning performance and student engagement. The employed algorithm is easy to implement and has low computational costs. In the experiment, we made use of only free and minimal intervention (email) and collaboration methods (email, VoIP, social media). Hence the organizational burden for developers, instructors and students was reduced to a minimum. The experiment is thus scalable and reproducible within many learning environments.

2. METHODOLOGY
2.1 Research Objectives
Empirically, we investigated the following three research questions:

1) Student engagement: Will MOOC students assigned to online groups (without further moderation) be engaged in online collaboration?

2) Drop-out rate: Will random or algorithmic grouping of MOOC students decrease the drop-out rate?

3) Learning performance: Can random or algorithmic grouping lead to higher learning performance, as measured in quizzes and homework scores?

2.2 Experiment Procedure
For conducting the experiment, we chose the second iteration of the course "The Fascination of Crystals and Symmetry", which was offered on the iversity.org platform. This is an introductory course to crystallography held by Dr. Frank Hoffmann (University of Hamburg). Since the course offered open discussion questions, it seemed well suited to engage students in group interaction. It had 3,209 enrollments in total, out of which 771 (i.e. 24.03%) were actively engaged throughout the course.

After the start of the course, 80 percent of the participants received a grouping survey via email asking for information about gender, timezone, language, personality, learning goals (general or in-depth) and their preferred collaboration method (local, email, Facebook, Google+ or Skype). The remaining 20 percent of the course received a motivational survey instead and served as a control condition. One week after the course start, students who provided sufficient answers to the grouping survey were assigned to groups of size 10 by our algorithm and received a second email a few days later. Those who did not respond but had a Facebook account were still randomly assigned to groups. The second email presented the other group members with their personal

* Zhilin Zheng and Tim Vogelsang contributed equally to this work.

descriptions as given in the survey. Further, the email contained a link to the first open discussion question of the course material and a link to their group (if applicable). Students from the control conditions, without or with insufficient grouping survey responses, were not assigned to groups. In addition, the course was held by Dr. Hoffmann as a flipped classroom at the University of Hamburg with approximately 65 students who watched the online lectures at home and met in-class for discussion. Out of these 65 students, 7 used their university account to sign-in to iversity and were anonymously included into our dataset. The other 58 students were not explicitly included. They either used private email addresses or did not sign up to the online course. This (relatively complex, but ecologically valid) assignment procedure of students to seven different conditions is summarized in Figure 1 and Table 1.

Table 1. Student conditions

Condition	Collaboration Method	Description
"Algorithm composed groups" (AlgoCG)	According to preference	Grouping survey, responded sufficiently grouped by algorithm
"Randomly composed groups" (RandCG)	Facebook	Grouping survey, not responded, Facebook user, grouped randomly
"Flipped classroom group" (FlippCG)	Local at University of Hamburg	Attended flipped classroom with the instructor
"No grouping - no answer" (NoG-NA)	none	Grouping survey, not responded, not grouped
"No grouping - insufficient answer" (NoG-IA)	none	Grouping survey, responded insufficiently, not grouped
"No grouping - control group - responsive" (NoG-CG-R)	none	Motivational survey, responded, not grouped
"No grouping - control group - nonresponsive" (NoG-CG-NR)	none	Motivational survey, not responded, not grouped

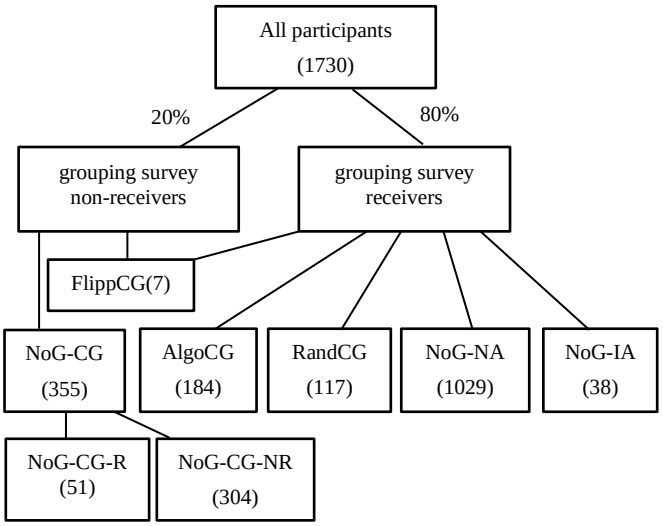


Figure 1. Student conditions with participation numbers.

As a last grouping related intervention, we sent a post grouping survey by the end of the course. This survey was only sent to the 80 percent who had also received the initial grouping survey and contained questions about satisfaction with and intensity of the group work.

3. GROUPING ALGORITHM

In order to create algorithm composed groups (AlgoCG), we used the collected responses from the grouping survey. We first segmented the respondents into five classes according to their collaboration preferences, namely local, email, Facebook, Google+ or Skype. For each class, we extracted each participant's gender, time zone, personality type, learning goal and language for the actual grouping. The task of the algorithm was to compose learning groups consisting of 10 students. Local groups were meant to only contain students from the same cities in order to actually meet up, resulting in very few and small groups qualifying for this option. The main algorithmic challenge was to take into account both heterogeneities (namely gender, personality type and learning goal) and homogeneities (i.e. time zone and language). Concretely, we wanted groups to have e.g. mixed gender, but similar time zone. To solve this optimization problem, we used a k-means clustering algorithm for fixed group sizes, based on [6]. The pseudocode of this algorithm is described in Figure 2 and our implementation in Python is publicly available[1]. In its original form, the algorithm calculates a homogeneity score for a single grouping criterion, like in usual applications of k-means clustering. For our experiment, we modified this algorithm to support multiple criteria and homogeneity as well as heterogeneity at the same time. As a modification, we calculated the group score as the difference between a homogeneity score (on time zone, language and learning goal) and a heterogeneity score (on gender and personality), both of which are actually measured by the Euclidean distance between peers.

```
---------------------------------------------------------------------
Step1: randomly assign students to groups;
Step2: for every group:
            for every student in the current group:
                calculate the possible group scores
                for the student in all the other groups;
                if the student has a higher group score
                in one of the other groups:
                    find the student in the other group with
                    the lowest group score;
                    swap the two students;
Step3: while we are significantly improving the average
        group score, go back to step 2
---------------------------------------------------------------------
```

Figure 2: K-means Clustering for fixed group sizes [6]. Image courtesy: Dirk Uys.

4. EXPERIMENT RESULTS

As a result of our grouping efforts, we composed 22 learning groups in total (4 local meeting groups, 5 Skype groups, 6 Facebook groups, 2 Google+ groups and 5 email groups). The

[1] *https://bitbucket.org/zhilinzheng7/kmeansgrouping*

following sub-sections present three aspects of the experiment results: student engagement, drop-out rate and learning performance.

4.1 Student Engagement

Roughly half of the students enrolled in the course were part of our experiment (1,730 out of 3,209). The other half of the students enrolled after the official start of the course (and, hence, after the start of our experiment), which is a usual pattern for a MOOC.

Overall, the course participants were quite inactive in general, as measured in terms of forum participation. Only 33 students participated in the forum by posting questions, answers or comments. The conducted post grouping survey had nine responses from participants that joined a group. Those respondents spent three hours on average (median one hour) on the group interactions. Further, the Facebook and Google+ groups that were created by us showed some initial greeting messages or comments but no deep, course-related interaction. Hence, our composition did not engage students in collaboration via the social media groups created for that purpose. For other online grouping participants (i.e. email and Skype) who did not answer to our post grouping survey, we cannot make the same conclusion owing to a lack of data.

However, students at least saw small descriptions of their peers in our welcome message and were partly able to see them on social media. Whether this fact, in addition to potentially unobserved interactions (e.g. via email), might have had an impact on the drop-out rate and learning performance, as well as how this relates to survey responsiveness, is analyzed in the following two subsections.

4.2 Drop-out Rate and Survey Responsiveness

We here define a 'drop-out' as any student who did not submit any quiz or assessment, and thereby did not qualify for any course score, after the group assignment.

Figure 3 shows the drop-out rate for all conditions. Unsurprisingly, all seven of the tracked flipped-classroom students stayed in the course (drop-out rate *0%*). In order to test the statistical significance of found differences in the drop-out rates, pairwise z-tests on the different conditions using a two-sided p-value were performed. For our conclusions about significance, we thus applied a Bonferroni correction to the significance level. The p-values in Section 4.2 are given in their non-Bonferroni-corrected form.

First of all, survey responsiveness plays a major role in the analysis. Among the participants of the treatment group that were not grouped, those who gave insufficient survey responses seem to be less likely to drop out than those who did not respond at all, yet this difference is not statistically significant (NoG-IA: *71.05%*, NoG-NA: *82.31%*, *p=0.07*). Further, in the control group without grouping, those who interacted with the motivational survey had a considerably lower drop-out rate than those who did not (NoG-CG-R: *62.75%*, NoG-CG-NR: *82.57%*, *p=0.001*). We can conclude that non-responsive students (with regard to a survey) are more likely to drop out than responsive students.

Hence, when analyzing the interplay between grouping condition and drop-out rate, we need to control for survey responsiveness. Since the randomly composed students did not respond to the grouping survey, we need to compare them to the students in the control group who did not respond to the motivational survey (RandCG: *77.78%*, NoG-CG-NR: *82.57%*, *p=0.26*). And since students from the algorithm composed groups responded to our grouping survey, they need to be compared with the fraction of the control group responding to the motivational survey (AlgoCG: *59.24%*, NoG-CG-R: *62.75%*, *p=0.65*). With this control for survey responsiveness, we thus find no statistically significant effects.

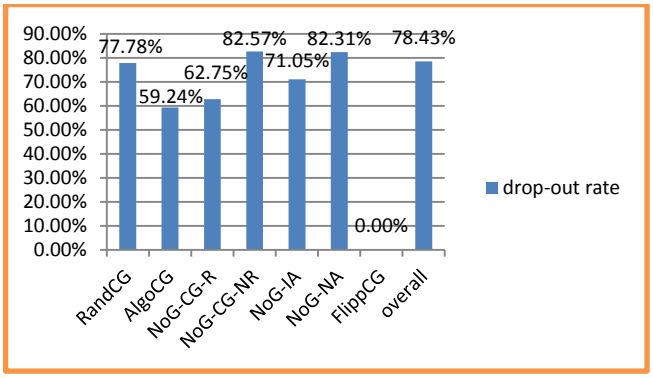


Figure 3: Bar-plot showing student's drop-out rates.

4.3 Learning Performance

In order to analyze the experiment's impact on student's learning performance, we looked at students' scores on quizzes and homework. Figure 4 visualizes average as well as minimum and maximum scores within the various experiment conditions. The flipped classroom condition outperformed all other conditions in terms of median score (FlippCG: 32, others: below 20). However, we do not find evidence for a positive impact of any condition on learning performance as measured by score. A one-way ANOVA implied no statistically significant difference between the conditions (*F(6,518)=1.284*, *p=0.265*).

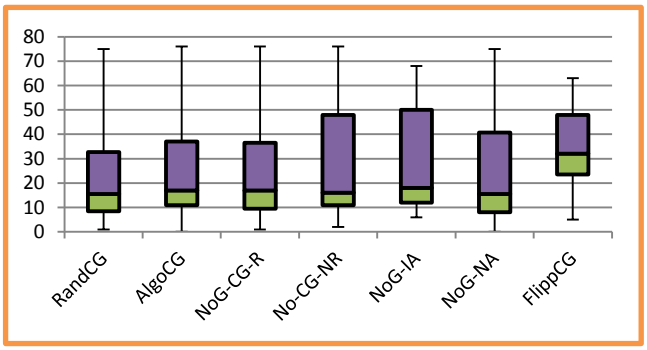


Figure 4. Boxplot showing student's scores from quizzes and homework.

5. DISCUSSION AND FUTURE WORK

In this paper, we presented a scalable and reproducible method to create small groups in online learning environments. We used minimal intervention methods and freely available collaboration tools as well as an adapted k-means clustering algorithm. Within the study, a flipped classroom approach outperformed all composed groups having no drop-out and above average learning scores. This is only partially surprising, as the flipped classroom students were in a formal education setting and most of the others were not. Further, survey responsiveness was found to be predictive of the drop-out rate. Comparing student conditions according to this insight, we found indications that composing

small learning groups in MOOCs (at least the way we did it) might not directly increase learning performance online, but could possibly decrease drop-out rates. However, these findings are limited by lack of statistical significance, self-selection biases and little observed interaction in the groups. These limitations need to be addressed within replications and extensions of this experiment.

Statistical significance: The scope of our experiment was a single but massive open online course with quite a high number of participants (1,730), which is far beyond the possibilities of a traditional classroom experiment. However, we faced low response rates and had to assign online students to rather complicated conditions, varying in size between 38 and 1,029 students (cf. Figure 1). The flipped classroom condition only had 7 students. Together, these impediments had a negative impact on the statistical power. For replication, even bigger courses should be chosen.

Self-selection: While only those who completed our grouping survey were assigned to the AlgoCG condition, we chose to compose RandCG from students who did not respond to this survey (for the sake of having enough groups in the AlgoCG condition). This self-selection problem was addressed analytically by also splitting our control group into responders and non-responders to our motivational survey. However, those interventions are not exactly equal: The email containing the motivational survey expresses the wish of the instructor and platform to get to know the students in order to adjust courses accordingly. The email containing the grouping survey, on the other hand, addresses the student's potential wish to collaborate in a group.

Group interaction: Finally, only very low actual collaboration could be observed in the Facebook and Google+ groups. How can small learning groups have an effect if nothing is going on in the groups? Some students claimed in the post grouping survey to have collaborated and it might be the case that the Facebook and Google+ groups were avoided (as an iversity team member was part of the group) and other, private, channels were preferred for collaboration.

In order to overcome the limitations within future student grouping experiments, we deduced new research hypotheses from our results.

Hypothesis 1: Using learning environments that are specifically designed for group work (including reminders, definition of learning goals, assignment of individual group roles or scheduled group meetings) will increase collaboration within small learning groups.

Hypothesis 2: Dynamic group (re-)composition using genetic or particle swarm algorithms will increase collaboration within the small learning groups, by solving the problem of drop-out in learning groups.

Hypothesis 3: Establishing small and regularly interacting sub-communities within a large online course may reduce students' drop-out rate. Just being aware of one another, even if not working together, is crucial.

6. ACKNOWLEDGMENTS

We thank Frank Hoffmann, Michael Sartor and Michael Fröba for collaborating with us and making this research possible; Jo Corrall for helpful feedback; and the China Scholarship Council (CSC) for providing a scholarship that supported this research.

7. REFERENCES

[1] COCEA, M. and MAGOULAS, G., 2010. Group Formation for Collaboration in Exploratory Learning Using Group Technology Techniques. In *Knowledge-Based and Intelligent Information and Engineering Systems*, R. SETCHI, I. JORDANOV, R. HOWLETT and L. JAIN Eds. Springer Berlin Heidelberg, 103-113.

[2] HUANG, Y.M., 2011. A systematic approach for learner group composition utilizing u-learning portfolio. *Journal of educational technology & society 14*, 3, 102-117.

[3] JORDAN, K., 2014. MOOC Completion Rates: The Data. In *Available at:http://www.katyjordan.com/MOOCproject.html*, [Accessed: 27/08/2014].

[4] OH, H., CHUNG, M.-H., and LABIANCA, G., 2004. Group Social Capital and Group Effectiveness: The Role of Informal Socializing Ties. *The Academy of Management Journal 47*, 6, 860-875.

[5] ONAH, D., SINCLAIR, J., and BOYATT, R., 2014. DROPOUT RATES OF MASSIVE OPEN ONLINE COURSES: BEHAVIOURAL PATTERNS. *EDULEARN14 Proceedings*, 5825-5834.

[6] UYS, D., 2014. How we Used the Echo Nest API for Engagement & Learning, Available at: http://info.p2pu.org/2014/01/13/how-we-used-the-echonest-api-for-engagement-learning/, [Accessed: May 2014].

[7] PAPPANO, L., 2012. The year of the mooc. In *In New York Times, November 2012*, Available at: http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html, [Accessed: October 2014].

[8] ROSÉ, C.P., CARLSON, R., YANG, D., WEN, M., RESNICK, L., GOLDMAN, P., and SHERER, J., 2014. Social factors that contribute to attrition in MOOCs. In *Proceedings of the first ACM conference on Learning @ scale conference* (Atlanta, Georgia, USA, 2014), ACM, 2567879, 197-198.

[9] ROSÉ, C.P. and SIEMENS, G., 2014. Shared task on prediction of dropout over time in massively open online courses. In *Proceedings of the 2014 Conference on Empirical Methods In Natural Language Processing Workshop on Modeling Large Scale Social Interaction In Massively Open Online Courses* (Doha, Qata, 2014), 39-41.

[10] SHIMAZOE, J. and ALDRICH, H., 2010. Group Work Can Be Gratifying: Understanding & Overcoming Resistance to Cooperative Learning. *College Teaching 58*, 2, 52-57.

[11] YANG, D., SINHA, T., ADAMSON, D., and ROSE, C.P., 2013. "Turn on, Tune in, Drop out": Anticipating student dropouts in Massive Open Online Courses. In *Proceedings of the Workshop on Data Driven Education, Advances in Neural Information Processing Systems* (2013).

Exploring Causal Mechanisms in a Randomized Effectiveness Trial of the Cognitive Tutor

Adam C Sales
Carnegie Mellon University
Pittsburgh, PA, USA
acsales@cmu.edu

John F Pane
RAND Corporation
Pittsburgh, PA, USA
jpane@rand.org

ABSTRACT

Cognitive Tutor Algebra I (CTAI), published by Carnegie Learning, Inc., is an Algebra I curriculum, including both textbook components and an automated, computer application that is designed to deliver individualized instruction to students. A recent randomized controlled effectiveness trial, found that CTAI increased students' test scores by about 0.2 standard deviations. However, the study raised a number of questions, in the form of evidence for treatment-effect-heterogeneity. The experiment generated student log-data from the computer application. This study attempts to use that data to shed light on CTAI's causal mechanisms, via principal stratification. Principal strata are categories of both treatment and control students according their potential CTAI usage; they allow researchers to estimate differences in treatment effect between usage subgroups. Importantly, randomization satisfies the principal stratification identification assumptions. We present the results of our first analyses here, following prior observational results. We find that students who encounter more than the median number of sections experience higher effects than their peers who encounter fewer, and students who need more assistance experience *lower* effects than their peers who require less.

Keywords

Causal Mechanisms, Principal Stratification, Intelligent Tutors, Bayesian Hierarchical Models

1. INTRODUCTION

The Cognitive Tutor Algebra I (CTAI) is a technology-based educational intervention that hopes to improve algebra I instruction by individualizing instruction to students needs, providing instant performance feedback, and implementing cognitive theories in mathematics education. [6]

Recently, a randomized controlled effectiveness trial, estimated the effect of a school's adoption of CTAI, under authentic conditions, on its students scores on an algebra proficiency exam. The results were reported in [4]. The study found that CTAI significantly increased test scores for 9th grade students in the second year of implementation, but was unable to detect effects in the experiment's first year, or in the 8th-grade group. These results raise a further question: by what mechanism, and for which students, does CTAI increase achievement? What usage patterns lead to higher effects? Can usage patterns explain the observed treatment effect heterogeneity?

The effectiveness trial produced extensive student usage data, as the computer program logged students' activity. In this paper, we begin use this data—in particular, usage data from the 2nd-year high school sample that apparently experienced a substantial CTAI effect—to explore the relationship between student usage and causal effects.

In doing so, we are guided by a previous study, [7] which (in one model specification) regressed post-test scores on CTAI usage variables, alongside student covariates and pre-test scores. That paper was aimed at post-test prediction, not causal inference, but it is of use in generating causal hypotheses: are there different effects for students who use CTAI for different amounts of time? Or for students who require more assistance from the program? Or for students who encounter more sections? This paper is a preliminary inquiry into these questions—more an exposition of the types of results that are possible than a full analysis—future work will delve more deeply into the data.

The data from the CTAI effectiveness study is invaluable for testing these hypotheses: due to its randomization design, we can draw causal conclusions without heroic assumptions. To do so, we will make use of the statistical framework of principal stratification, which we will describe in the following section. The next section will describe our models in detail, and results and conclusions will follow.

2. PRINCIPAL STRATIFICATION

Following [8], we conceptualize causal inference in terms of counterfactuals: comparing what students would have experienced with CTAI with what they would have experienced in its absence. In particular, if Y is the outcome of interest, in our case, post-test scores, we may define two "potential outcomes" for each subject: $Y_i(0)$ is what a subject i would score on the post-test if i's school were assigned to the control condition, and $Y_i(1)$ is what I would score if her school were assigned to treatment.

Principal stratification (PS) [2] is an approach to modeling a categorical or discrete post-treatment variable M within the potential outcomes framework. When treatment assignment Z is binary, each subject i has two potential values of M: $M_i(0)$—the value of M that would be observed under the control condition—and $M_i(1)$, what would be observed under the treatment condition. These define subgroups—principal strata—within which causal effects may be defined. In particular, a principal causal effect is

$$Y(1) - Y(0)|M(1) = m, M(0) = m' \quad (1)$$

that is, the effect of Z on Y among those subjects with particular potential outcomes for M of m and m'.

In this study, following [3], we use principal stratification to examine some hypothesized causal mechanisms of CTAI. For instance, consider the usage variable $totalTime$: the total amount of time students spend working CTAI problems. Since $totalTime$ is continuous, we begin by dichotomizing it; for the sake of simplicity, let $\mu = median(totalTime)$ and $M = \mathbb{1}_{[totalTime>\mu]}$. We can define four principal strata. The first is comprised of those students who, if assigned to CTAI, would use it for more time than μ—$M(1) = 1$—but if assigned to the control condition would use it less, $M(0) = 0$. Next, consider the group $M(1) = 0$; $M(0) = 1$, those students who use CTAI for less time because of their treatment assignments. The remaining two groups are $M(1) = 0$; $M(0) = 0$ and $M(1) = 1$; $M(0) = 1$, those students who would use CTAI less for less, or more, time than m regardless of treatment assignment. By examining differences between the average treatment effects in the four groups, we can learn how CTAI's impact varies for different usage patterns.

Randomization allows us to estimate principal effects as the average treatment minus control difference in gain scores within each estimated stratum. That is, randomization of treatment assignment leads to identification of principal effects: the effect of Z within principal strata. On the other hand, the difference in treatment effects between principal strata does not necessarily estimate a causal quantity. Randomization does not identify students' counterfactual gain scores had they been in alternative principal strata. That being said, differences in treatment effects across strata can suggest causal mechanisms.

Fortunately, the CTAI study's design substantially simplifies the PS analysis, by eliminating two of the principal strata. Students in the control group had (for the most part) no access to the CTAI program. Therefore, we can safely assume that for all students, $M(0) = 0$. This leaves two principal strata, $M(0) = 0$; $M(1) = 0$, and $M(0) = 0$; $M(1) = 1$—that is, the students who, if assigned to treatment, would use CTAI for more time than m and those who would not. Only one of the potential values of M is directly observed; in particular, $M(1)$ is unknown for subjects in the control group. Stated differently, the values $M(1)$ are missing for students in the control group, but they may be imputed because the "missingness mechanism," treatment assignment, is random, or ignorable. Therefore, randomization of treatment assignment allows us to identify members of each principal stratum, and effects of treatment within those strata.

3. MODELING STRATA AND OUTCOMES

In this preliminary study, we considered three of the usage variables previously modeled as predictors in [7]: $totalTime$, the total amount of time students spent working CTAI problems, $numSec$, the number of sections each student encountered, and $assistance$, the average sum of hints and errors per problem for each student. We ran a separate PS model for each usage variable, but all three PS models had the same form. Each PS model itself was a combination of two multilevel models. The first, fit only within the treatment group, modeled the usage variable M as a function of covariates X_t. This model was used to estimate the usage that control students would have experienced had they been assigned to treatment. The second model used the results of the first model, and a somewhat larger set of covariates X_y, to estimate the effect of random assignment to CTAI in each of the principal strata.

3.1 Usage Model

Modeling each usage variable was a four-step process: first, we calculated the variable's values from the available data; next, we transformed those values so that their observed distributions would be closer to a normal distribution; next, we modeled the transformed variables as a linear function of covariates X_t, and finally, we dichotomized the model's output, to define and estimate principal strata.

As students used CTAI, the program recorded timestamps at the beginning and end of each problem. The difference between these two is the amount of time the student spent on each problem, recorded in milliseconds. The sum of was the variable $totalTime$. The distribution of $totalTime$ was heavily skewed rightward, so we transformed it to ease the modeling process. The transformation that resulted in a distribution whose histogram appeared approximately normal was a box-cox transformation with a parameter of 0.3 [1].

Next, we modeled the transformed $totalTime$ as a function of a set of covariates X_t containing dummy variables for the state in which the school was located, the student's grade, race, sex, special education status, free or reduced-price lunch status and pretest scores, along with missingness indicators. Formally, the model was

$$totalTime_{ijk} = \alpha + X_{ti}^T\beta + \epsilon_{ijk} + \eta_{jk} + \nu_k \quad (2)$$

where α and β are, respectively, an intercept and a vector of coefficients estimated from the data, $\epsilon_{ijk} \sim N(0, \sigma_{st})$ is a student-level random error, $\eta_{jk} \sim N(0, \sigma_{tt})$ is a random effect for teacher, and $\nu \sim N(0, \sigma_{ut})$ is a random effect for school. The variance parameters σ_{st}, σ_{tt} and σ_{ut} are estimated from the data. In other words, $totalTime$ was modeled as multilevel, with students nested within teachers, nested within schools.

The transformed $totalTime$ values, or, in the case of the control sample, their predictions, gave rise to a dichotomous variable M, which took the value of 1 if $totalTime$ or its prediction is greater than its observed median of about 22 hours over the course of the year. The variable M defined two principal strata: those students with $M(1) = 1$ and those with $M(1) = 0$.

CTAI also automatically collected data on the number of

hints and errors students request or make. Following [7], we normalized hints and errors by section. Next, we averaged the normalized values by student, producing average assistance per problem, or *assistance*. We transformed *assistance* in the same was as *totalTime*. Next, we modeled *assistance* with equation (2), and dichotomized the results using their observed median, 0.076, which, due to the prior normalization, is not a whole number.

The third usage variable we considered here is *numSec*, the number of sections students encountered on CTAI. We transformed *numSec* with a natural logarithm, modeled it with equation (2), and dichotomized it with its median, 27 sections.

3.2 Outcome Model

For each dichotomized usage variable M, we fit a multilevel linear model to estimate principal effects of CTAI treatment on post-test scores. The post-test from the CTAI effectiveness study is the Algebra Proficiency Exam. It was analyzed with item-response-theory, and its reported scores have a mean of 0 and a standard deviation of 1, so regression coefficients may be interpreted as effect sizes [4]. To account for pre-test scores, while avoiding measurement-error concerns, we modeled students' gain scores, $\texttt{diff}_{ijk}$, the difference between their post-test and pre-test scores. The student-level model, then, was

$$\begin{aligned}\texttt{diff}_{ijkm} =& \alpha' + X_{yi}^T\gamma + \lambda M_{ijkm} + \tau Z_{km} \\ &+ \kappa Z_{km} M_{ijkm} + \epsilon'_{ijkm} + \eta'_{jkm} \\ &+ \nu'_{km} + \zeta_m \end{aligned} \quad (3)$$

Here α', ϵ', η', and ν' are, respectively, an intercept, and random effects for individual, teacher, and school. The apostrophes indicate that these are distinct from their analogues in equation (2). There is an additional random effect ζ for "match," accounting for the matched-pair randomization design. X_t is a vector of covariates equivalent to those in (2), with the addition of standardized test scores from the prior two years. The principal effects emerge from the coefficients τ and κ: τ is the average effect in the $M(1) = 0$ group, and $\tau + \kappa$ is the average effect in the $M(1) = 1$ group. Finally, λ is the difference in $Y(0)$ between the $M(1) = 1$ and $M(1) = 0$ groups.

Models (2) and (3) were fit simultaneously in JAGS [5], a Bayesian Gibbs sampler. To facilitate Bayesian model fitting, we provided weakly informative priors on all of the model parameters.

4. RESULTS

	Estimate	SE	95% Interval
$M(1) = 0$	0.12	0.06	(0.01,0.25)*
$M(1) = 1$	0.32	0.27	(-0.21,0.85)
Difference	0.20	0.27	(-0.32,0.74)

Table 1: Results for *totalTime*. Point estimates for effect size, standard errors and 95% credible intervals for the average treatment effects in two principal strata, denoted $M(1) = 1$ and $M(1) = 0$, as well as the difference between the two.

We present results for each of the three usage variables we considered. For each variable, we present the average treatment effect for subjects in the $M(1) = 0$ stratum—that is, students whose usage under the treatment condition was, or would be, less than the observed median—the effect for students in the $M(1) = 1$ stratum, and the difference between the two effects. For each effect, we present a point estimate, equivalent to the mean of the posterior distribution, a standard error—the standard deviation of the posterior—and a 95% credible interval, representing the 0.0275 and 0.975 quantiles of the posterior. Effects whose credible interval does not include 0 are marked with an asterisk.

Like [7], we were unable to establish that students who spend more time using CTAI gain more from its use. The relevant results are available in Table 1. We estimated an effect size of 0.12 in the low-time group $M(1) = 0$, and 0.32 in the high-time group $M(1) = 1$. However, the standard errors were too large to draw strong conclusions.

	Estimate	SE	95% Interval
$M(1) = 0$	-0.02	0.07	(-0.16,0.11)
$M(1) = 1$	0.30	0.13	(0.13,0.47)*
Difference	0.32	0.09	(0.14,0.49)*

Table 2: Results for *numSec*

On the other hand, as seen in Table 2, students who encountered a greater number of sections (or would have, had they been assigned to treatment) experienced a much larger effect than those who encountered fewer sections. The effect size for students who encounter more than the median number of sections is, with 0.95 probability, between 0.13 and 0.47—a very large effect. This is about 0.32 higher than for the students who encountered fewer sections, for whom there was no discernible effect at all.

	Estimate	SE	95% Interval
$M(1) = 0$	0.30	0.08	(0.14,0.45)*
$M(1) = 1$	0.12	0.09	(-0.07,0.30)
$M(1) = 2$	-0.08	0.08	(-0.23,0.06)
Difference (0–1)	-0.18	0.09	(-0.34,-0.00)*
Difference (1–2)	-0.20	0.09	(-0.36,-0.01)*

Table 3: Results for *assistance*

Lastly, we suspected that the relationship between assistance and CTAI effect might not be monotonic. That is, it might be that the effect of CTAI is low for students who request many hints and make many errors, but high for those with a medium amount, or vice versa. For that reason, we split the variable at the 1/3 and 2/3 quantiles, and estimated three principal effects. Our suspicion proved false, however, and the result was similar to what was reported in [7]: higher hints and errors corresponded to lower CTAI effects. For students who requested few hints and made few errors, the effect was between 0.14 and 0.45, while 95% intervals for the other two strata included 0. Ninety-five percent intervals on the difference from one strata two the next were entirely negative.

5. DISCUSSION

This work was a first look at causal modeling with usage variables from a randomized experiment of educational soft-

ware. We showed that without additional identification assumptions, researchers can use log data to form a deeper understanding of their software's effect. That being said, this work is preliminary, both because the statistical models we used may be improved, and because much more information is available in the CTAI log data.

In this paper, we focused on three hypotheses that were suggested in [7]. That paper used a linear regression model, fit using a convenience sample of CTAI users, to show that certain usage variables, among which are the total amount of time students spend solving problems, the number of sections students encounter, and the assistance the software provides them, can predict standardized test scores, even after controlling for a number of baseline covariates. With some very strong assumptions, one may interpret [7]'s results as causal: that seeing more sections, for instance, causes students to achieve higher test scores.

In our design, by contrast, the estimated treatment effects—comparisons between treatment and control students—are inherently causal due to the randomization design. The principal stratification approach allows us to reliably estimate causal effects within the strata. That said, this approach largely replicates the results from [7]. Students who spend more time working on CTAI problems seem to experience a larger effect, but this conclusion is ultimately unclear: the credible interval of the difference in effects between students who use the program for more time and those who use it for less contains 0. On the other hand, we found that students who encounter more sections do indeed experience larger effects. One reason for this result may be that the effect a CTAI user feels is particular to the skills the user practices—students who encounter a wider array of sections learn more from CTAI, and their performance on a wider array of sections of the posttest is improved. At the same time, students who required more assistance per problem—that is, asked for more hints and made more errors—experienced a smaller effect than their peers who required less assistance. This may be for a number of reasons. For instance, perhaps students who need more assistance per problem are struggling more, and have a greater need for a teacher's help. Alternatively, students who ask for a lot of hints and make a lot of mistakes may not be trying their hardest on CTAI, and for that reason may not experience the same rewards from CTAI. More research and data analysis is necessary to properly interpret these results.

Along those lines, we plan a number of future analyses. First, improved models may help us understand the relationships that this paper explores. For instance, dividing the usage variables into three or more categories may be more illuminating than the two categories we explore here. Additionally, it may be useful to match section- or unit-specific usage to appropriate items on the posttest.

Further along, we hope to discover and define interesting multivariate principal strata, perhaps as the result of a cluster analysis of the high-dimensional usage data.

Finally, after cultivating a more complete understanding of the usage patterns that lead to higher CTAI effects, we can explore treatment-effect heterogeneity. In particular, we may be able to answer why in the first year of implementation CTAI did not seem to boost test scores, but in the second year it did. Was differential usage to blame?

In the meantime, this paper uses rigorous causal methods to confirm some previous hypotheses about CTAI's causal mechanisms, and points a way forward for future work modeling usage variables in experimental designs.

6. ACKNOWLEDGMENTS

This work is supported by the United States National Science Foundation Grant #DRL-1420374 to the RAND Corporation and by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B1000012 to Carnegie Mellon University. The opinions expressed are those of the authors and are not intended to represent views of the Institute or the U.S. Department of Education or the National Science Foundation. Thanks to Steve Fancsali, Steve Ritter, and Susan Berman for processing and delivering the CTAI usage data.

References

[1] G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[2] C. E. Frangakis and D. B. Rubin. Principal stratification in causal inference. *Biometrics*, 58:21–29, 2002.

[3] L. C. Page. Principal stratification as a framework for investigating mediational processes in experimental settings. *Journal of Research on Educational Effectiveness*, 5(3):215–244, 2012.

[4] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 2013.

[5] M. Plummer. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 2003.

[6] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255, 2007.

[7] S. Ritter, A. Joshi, S. E. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *Proc. of the 6 th International Conf. on Educational Data Mining*, 2013.

[8] D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology; Journal of Educational Psychology*, 66(5):688, 1974.

Confounding Carelessness? Exploring Causal Relationships Between Carelessness, Affect, Behavior, and Learning in Cognitive Tutor Algebra

Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, 20th Floor
Pittsburgh, PA 15219 USA
1.888.751.8094 x219
sfancsali@carnegielearning.com

ABSTRACT

Studies have found positive correlations between affective states (e.g., confusion, boredom) and learning outcomes in educational technologies like ASSISTments and Carnegie Learning's Cognitive Tutor. The adage that "correlation does not imply causation" is especially apt in light of these observations; it seems counterintuitive that increasing student boredom or confusion (e.g., designing systems that bore or confuse students) will benefit learning. One hypothesis to explain positive correlations between boredom and learning suggests that carelessness is a "confounding" common cause of boredom and another construct linked to learning. We consider a Cognitive Tutor Algebra dataset in which boredom and confusion are positively correlated with learning. Prior causal modeling of this data suggests that various behavioral and affective features (e.g., boredom and gaming the system) share unmeasured common causes. We provide a correlational analysis and causal models of this data that situate carelessness among behaviors and affective states to determine whether (and how) carelessness plays a confounding role.

Keywords

causal models, causal discovery, structural equation modeling, carelessness, boredom, confusion, affect, gaming the system, off-task behavior, Cognitive Tutor, intelligent tutoring systems

1. INTRODUCTION

Recent research in educational data mining has led to the development of sensor-free, data-driven approaches to "detect" various behavioral and affective features from logs of learner interactions with technologies like intelligent tutoring systems (ITSs). Since such approaches to detecting phenomena like "gaming the system" [3-4], off-task behavior [1], and affective states [5] have been validated against field observations of learner behavior, a natural next step for researchers has been to use the predictions of detectors as inputs to predictive models of substantive learning outcomes in what have been called "discovery with models" approaches [6]. Such approaches have sought to answer questions about whether the tendency of learners to game the system or become bored using a system are predictive of outcomes like post-tests and standardized test scores (e.g., [11, 17]).

Our recent work [13] advocates seeking causal knowledge about learner behavior, affect, and learning, even when faced with non-experimental data, and that graphical causal models and data-driven search for their structure [22] provide an avenue for *causal* discovery with models. Findings using data from Carnegie Learning's Cognitive Tutor (CT) ITS [19] suggested that most affect and behavior variables shared unmeasured common causes.

The present work integrates detectors of carelessness into this work [13]. Carelessness was correlated with a variety of affective phenomena in an ITS with features similar to CT (e.g., [21]) and has been hypothesized to play a causal role among affective states as well (e.g., as a cause of boredom [17] or effect of boredom [8]). Other work emphasizes relationships between engagement and carelessness [9-10]. Our findings suggest a causal link between concentration and carelessness and possible causal links between confusion, gaming the system, and carelessness.

2. PRELIMINARIES

2.1 Motivation & Outline

Recent studies (e.g., [13, 17]) observe positive correlations between learning and the propensity to be in affective states like boredom and confusion, but it seems counterintuitive that increasing student boredom or confusion is likely to benefit learning (i.e., that these correlations are because of causal links). Several hypotheses have been proffered to explain such positive correlations. One hypothesis, for the ASSISTments system [14], is that learners become bored when they make careless mistakes and are required to work through step-by-step breakdowns of math problems [17]; learners with greater knowledge are more likely to be careless and bored, but since they are capable learners they will have better learning outcomes, providing a possible explanation for a positive correlation between boredom and learning.

Further, causal models of affect, behavior and learning in CT Algebra finds that boredom and gaming the system behavior are negatively correlated and suggest that they share an unmeasured (or latent) common cause (i.e., a "confounding" variable) [13]. Boredom's negative correlation with gaming the system, and gaming behavior's negative correlation with learning helped to explain the overall positive correlation of boredom and learning in that study. This same study also found a positive correlation between confusion and learning, and causal models suggested that confusion and gaming may be confounded. The hypothesis of [17] about carelessness may be appropriate for CT; incorrect responses despite knowledge will lead students to be presented more practice on skills they already know because CT will decrease its

estimate of skill mastery based on incorrect responses, and this could lead to boredom.

Recent work has focused on modeling carelessness [20] in systems like ITSs by using context-sensitive models to predict whether particular incorrect responses are likely examples of "slips" in which students answer incorrectly despite knowing a skill [2], and have explored correlations between carelessness and affective states (e.g., [21]).

We now introduce CT Algebra and detector models. We then review graphical causal models and data-driven structure search before explaining prior work and presenting novel causal models that incorporate carelessness. We conclude with discussion.

2.2 Cognitive Tutor (CT) Algebra

Carnegie Learning's CT is an ITS for mathematics used by hundreds of thousands of learners every year across the United States and internationally. CT breaks down mathematics subject areas like algebra into fine-grained skills or knowledge components (KCs), the mastery of which is used to determine learner progress through a series of topical sections that comprise broader units. Each section is comprised of multi-step problems that allow for the assessment of student progress toward mastery of fine-grained KCs.

CT assesses KC mastery using a probabilistic framework called Bayesian Knowledge Tracing (BKT) [12]. BKT assesses learner progress to mastery by assuming that a learner is either in the "unknown" state for a KC or the "known" state for a KC (i.e., KC mastery) and uses observations of practice opportunities for each KC to predict the state of a learner is at any given time. To make this prediction, BKT provides for four parameters for each KC: (1) the probability of prior knowledge or mastery of the KC, (2) the probability of a transition from the unknown to the known state at a given KC practice opportunity, (3) the probability that a learner guesses (i.e., is in the unknown state but answers correctly), and (4) the probability that the learner "slips" (i.e., has mastered a KC but provides an incorrect response).

2.3 Affect, Behavior, & Carelessness

Educational data mining researchers seek to avoid obtrusive, costly, and non-scalable sensor-based methods for measuring learner (dis-) engagement and affect with systems like ITSs by developing data-driven predictive models, frequently referred to as "detectors" that rely only on features that can be "distilled" from fine-grained log data. Detector models use machine learning methods applied to distilled features to make predictions about whether particular learner interactions with a system are likely to be instances of particular types of behavior. Detector models are validated against field observations in real classrooms. For correlational and causal modeling, we quantify levels of behavior per student by calculating the proportions of problem-solving steps deemed to be likely the result of behaviors like gaming the system or off-task behavior, which we now briefly explicate.

Gaming the system [3-4] refers to behavior in which learners attempt to make progress through content without genuinely learning or mastering appropriate skills (e.g., by incorrectly providing numbers within problem statements). A robust finding of previous efforts is that there is evidence that gaming the system is a cause of decreased learning [13]. Off-task behavior refers to learner disengagement from the learning environment and learning [1]. Recent efforts did not find evidence for a causal link between off-task behavior and learning.

Evidence also suggests that affective states play an important role in learning (e.g., [18]). Detector models similar to those for gaming the system and off-task behavior have been developed for affective states like boredom, confusion, and engaged concentration [5]. Modeling efforts for a CT Algebra dataset provided a somewhat complicated causal picture; while boredom and confusion may be *negatively* correlated with another factor that causes *decreased* learning, gaming the system, (hence positively correlated with learning), there are likely unmeasured common causes of these states and gaming the system.

Learner carelessness has been discussed as problematic in classrooms since at least the 1950s [21]. Other work identifies carelessness as a problem even among high-performing students [9-10]. Recent work on data-driven detector models [20] seeks to operationalize carelessness by focusing on the notion of "slipping," when learners answer incorrectly despite knowing a skill. In standard BKT, the parameter for slipping remains constant per KC over time; contextual models of guessing and slipping predict whether particular correct and incorrect responses are likely the result of learners guessing or slipping based on aspects of their performance [2]. The contextual slip model that predicts whether particular incorrect responses are instances of slipping is built in the same manner as other detector models. Operationalized as contextual slipping, carelessness can be quantified on a per learner basis by calculating the mean probability with which contextual slip models predict that incorrect actions are examples of slipping [21].

2.4 Graphical Causal Models & Model Search

We adopt directed acyclic graphs (DAGs) to represent causal relationships among variables we seek to model. We consider the context of linear relations among variables and multi-variate Gaussian joint probability distributions, where DAGs imply conditional independence constraints on observed joint distributions and covariance matrices. The set of DAGs consistent with a set of independence constraints, assuming that there are no unmeasured common causes of measured variables, comprise an equivalence class of graphs, represented by a graphical object called a pattern. Patterns and other equivalence classes of graphs can be inferred from data by asymptotically reliable algorithms developed (e.g., the TETRAD[1] project) over the past 20+ years.

We deploy the constraint-based PC algorithm to learn a pattern from data, making the strong assumption of no unmeasured common causes of measured variables [22]. From a pattern, we can choose a DAG member of the equivalence class to specify a linear structural equation model (SEM). Allowing for unmeasured common causes, we consider an equivalence class of graphs, represented by Partial Ancestral Graphs (PAGs), learned using the FCI algorithm [22]. FCI is similar to PC, but PAGs have a richer set of edges between two variables X and Y in a PAG [13, 22]:

- X o—o Y: (1) X is an ancestor (i.e., cause) of Y; (2) Y is a cause of X; (3) X and Y share a latent common cause; (4) either (1) & (3) or (2) & (3).
- X o→ Y: Either X is a cause of Y; X and Y share a latent common cause; or both.
- $X \leftrightarrow Y$: X and Y share a latent common cause in every member of the equivalence class represented by this PAG.

[1] freely-available at http://www.phil.cmu.edu/projects/tetrad/

- $X \rightarrow Y$: X is an ancestor/cause of Y in every member of the equivalence class represented by this PAG.

3. DATA + PRIOR WORK

Our data are logs for a sample of 102 adult, higher education learners using CT Algebra. We consider log data over roughly 337,000 learner actions in a module of five units concerning linear equations and inequalities, relatively late in the course. We also have a pre-test score (*Module Pre-Test*) and a *Final Exam* score for the entire algebra course, which is our learning outcome.

Assuming no unmeasured common causes of variables, causal models of this data [13] illuminated one possible explanation for the positive correlations between both *Boredom* and *Confusion* and *Final Exam*: both may cause decreased *Gaming the System* behavior, behavior which is found to cause decreased learning. While *Confusion* may cause decreased *Gaming the System* (e.g., *Confusion* being an affective state in which learners are unlikely to be able to "game"), there are reasons to suspect that this correlation and others arise due to confounding common causes.

Relaxing the assumption of no unmeasured common causes and allowing affect and behavior to co-occur, the FCI algorithm found a *robust* causal link between gaming and learning; all other links in the PAG causal model from prior work are at least possibly confounded. This fact and several common cause hypotheses in the literature explaining positive links between *Confusion* and *Boredom* and learning lead us to consider *Carelessness*.

4. MODELING CARELESSNESS

4.1 Correlational Analysis

Carelessness is positively correlated with both *Module Pre-Test* ($r = 0.36$, $p < .001$) and *Final Exam* ($r = 0.56$, $p < .001$), consistent with results that careless behavior is common even among high-performing math learners [9-10]. Correlations of *Carelessness* to other affective and behavioral variables are presented in Table 1. These results are largely consistent with those in previous work analyzing the relationship between *Carelessness* and affect [21].

Table 1. Pairwise correlations of *Carelessness* and other variables representing "detected" behavior and affective states (*p < .05; ***p < .001)

Variable / Construct	Pearson Correlation
Boredom	0.13
Confusion	0.48***
Engaged Concentration	0.75***
Gaming the System	-0.74***
Off-Task Behavior	-0.25*

4.2 Causal Models

Rather than attempt to specify and test "by hand" a multitude of alternative models that posit different causal roles for *Carelessness*, we adopt a search strategy. Assuming that affective states (including *Carelessness*) causally precede behavioral variables, the PC algorithm learns the DAG causal structure of the estimated linear SEM of Figure 1. This model fits the data ($\chi^2(19) = 23$; $p = .22$) [7] and is similar to that the model found in previous work under the same assumptions [13]. We focus on three elements of it.

First, *Engaged Concentration* is inferred to be a cause of *Carelessness*, consistent with the high correlation in the Scatterplot Study [21], and hypotheses due to Clements [10] about the relationship between engagement (i.e., *Engaged Concentration*) and *Carelessness*. San Pedro, et al. note the positive link between confidence and *Carelessness* found by Clements and posit that an engaged learner of only average knowledge might become overly confident in their ability and careless [16, 21]. This explanation suggests an intermediary along this causal pathway, a topic for future research.

Second, *Carelessness* is inferred to be a common cause of *Confusion* and *Gaming the System*, with increased *Carelessness* leading to increased *Confusion* and less *Gaming the System*. *Carelessness* as a common cause of these two variables is consistent with models in [13] in which an edge *Confusion* $\circ\!\!\rightarrow$ *Gaming the System* indicated the possible presence of an unmeasured (i.e, confounding) common cause. The strong positive relationship between *Engaged Concentration* and the inferred cause of *Confusion*, *Carelessness*, provides a plausible explanation for the positive correlation of *Confusion* and learning, but this model does not suggest we pursue interventions that increase learner *Confusion*, though recent literature suggests that, in some contexts, *Confusion* may be beneficial for learning (e.g., [15]).

With respect to the other effect of *Carelessness* in Figure 1, *Gaming the System*, it is possible that there is a negative causal connection, as presumably gaming behavior is the result of at least a certain amount of non-careless affect and corresponding behavior, as learners must provide roughly appropriate responses to math problems if they are to, in fact, "game the system." However, it is also plausible that *Carelessness* and *Gaming the System* share a confounding common cause.

Relaxing the assumption of no unmeasured common causes and assuming only that *Module Pre-Test* precedes all affective and behavioral variables, all of which precede *Final Exam*, FCI learns the PAG causal model in Figure 2, with +/- signs to remind the reader of parameter estimates in Figure 1. Contrary to the model of Figure 1, either *Confusion* is a cause of *Carelessness*, or they share an unmeasured common cause. The direction of the link between *Confusion* and *Carelessness* is sensitive to the "ordering" of affective and behavioral variables. However, under nearly all combinations of behavioral and affective variable orderings and groupings, *Engaged Concentration* is a cause of *Carelessness*, consistent with past hypotheses [9-10] and correlational analyses [21]. While we infer that *Carelessness* and *Gaming the System* share an unmeasured common cause, relationships between variables like *Carelessness* and *Gaming the System* may be confounded, not only by other unmeasured phenomena, but by the underlying phenomenon itself since we provide only noisy measures using detector models.

5. DISCUSSION

We provide evidence for the hypothesis that concentration leads to (i.e., causes) careless mistakes, and this causal inference is robust under a variety of assumptions. Contrary to some hypotheses [8, 17], we do not find evidence for a causal link between *Carelessness* and *Boredom* in CT Algebra. However, that hypothesis of [17] was made with respect to the ASSISTments system. Future research should take on the problem of learning causal models from available observational data from systems like CT and ASSISTments to determine under what circumstances causal inferences of the sort we consider here generalize across

sub-populations using the same instructional system as well as different systems within the same (or different) domain.

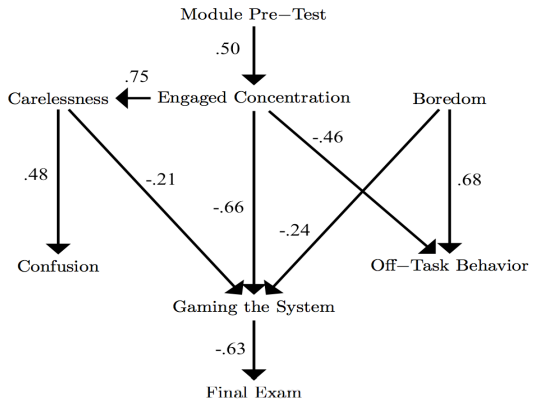


Figure 1. Estimated SEM incorporating *Carelessness*

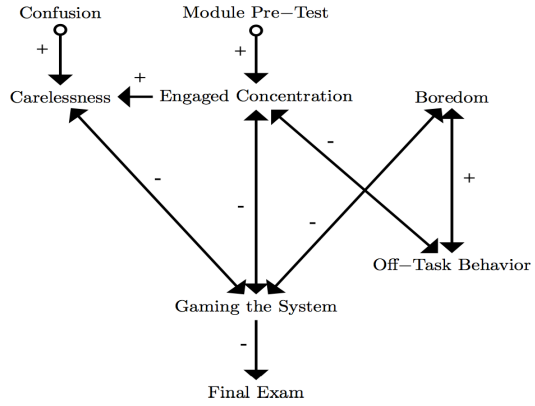


Figure 2. PAG causal model incorporating *Carelessness*

6. ACKNOWLEDGMENTS

The author gratefully acknowledges Ryan S.J.d. Baker, Susan R. Berman, Ryan Carlson, Sujith M. Gowda, Steven Ritter, and Charles Shoopak for providing code, assistance, and/or comments.

7. REFERENCES

[1] Baker, R.S.J.d. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proc. of ACM CHI 2007: Computer-Human Interaction* (San Jose, CA, April 28 – May 3, 2007). ACM, New York, 1059-1068.

[2] Baker, R.S.J.d., Corbett, A.T., Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian Knowledge Tracing. In *Proc. of ITS 2008* (Montreal, Canada, 2008). 406-415.

[3] Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R. 2008. Developing a generalizable detector of when students game the system. *User Model. User-Adap.* 18 (2008), 287-314.

[4] Baker, R.S.J.d., de Carvalho, A. M. J. A. 2008. Labeling student behavior faster and more precisely with text replays. In *Proc. of EDM 2008* (Montreal, 2008). 38-47.

[5] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., Rossi, L. 2012. Towards sensor-free affect detection in Cognitive Tutor Algebra. In *Proc. of EDM 2012* (Chania, Greece, 2012). 126-133.

[6] Baker, R.S.J.d., Yacef, K. 2009. The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining* 1 (2009), 3-17.

[7] Bollen, K. 1989. *Structural Equations with Latent Variables*. John Wiley & Sons.

[8] Cheyne, J.A., Carriere, J.S., Smilek, D. 2006. Absent-mindedness: lapses of conscious awareness and everyday cognitive failures. *Conscious Cogn* 15 (2006), 578-592.

[9] Clements, M.A. 1980. Analyzing children's errors on written mathematical tasks. *Educational Studies in Mathematics* 11, (Feb. 1980), 1-21.

[10] Clements, M.A, 1982. Careless errors made by sixth-grade children on written mathematical tasks. *Journal for Research in Mathematics Education* 13, (Mar. 1982), 136-144.

[11] Cocea, M., Hershkovitz, A., Baker, R.S.J.d. 2009. The impact of off-task and gaming behavior on learning: immediate or aggregate? In *Proc. of AIED 2009* (Brighton, UK, 2009). 507-514.

[12] Corbett, A.T., Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap.* 4 (1995), 253-278.

[13] Fancsali, S.E. 2014. Causal discovery with models: behavior, affect, and learning in Cognitive Tutor Algebra. In *Proc. of EDM 2014* (London, UK, 2014). 28-35.

[14] Feng, M., Heffernan, N.T., Koedinger, K.R. 2009. Addressing the assessment challenge in an intelligent tutoring system that tutors as it assesses. *User Model. User-Adap.* 19 (2009), 243-266.

[15] Lehman, B., D'Mello, S.K., Graesser, A.C. 2012. Confusion and complex learning during interactions with computer learning environments. *Internet High. Educ.* 15 (2012), 184-194.

[16] Linnenbrink, E.A., Pintrich, P.R. 2003. The role of self-efficacy beliefs in student engagement and learning in the classroom. *Reading & Writing Quarterly: Overcoming Learning Difficulties* 19 (2003), 119-137.

[17] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. 2014. Affective states and state tests: investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics* 1 (2014), 107-128.

[18] Pekrun, R., Goetz, T., Titz, W., Perry, R.P. 2002. Academic emotions in students' self-regulated learning and achievement: a program of quantitative and qualitative research. *Educ. Psychol.* 37 (2002), 91-106.

[19] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14 (2007), 249-255.

[20] San Pedro, M.O.C.Z., Baker, R. S. J. d., Rodrigo, M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *Proc. of AIED 2011* (Auckland, New Zealand). 304-311.

[21] San Pedro, M.O.C.Z., Baker, R.S.J.d., Rodrigo, M.T. 2014. Carelessness and affect in an intelligent tutoring system for mathematics. *Int J Artif Intell Educ* 24 (2014), 189-210.

[22] Spirtes, P., Glymour, C., Scheines, R. 2000. *Causation, Prediction, and Search*. 2nd Edition. MIT, Cambridge, MA.

Students at Risk: Detection and Remediation

Irena Koprinska, Joshua Stretton and Kalina Yacef
School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia
{irena.koprinska, joshua.stretton, kalina.yacef}@sydney.edu.au

ABSTRACT
Detecting students at risk of failing is particularly useful and desirable when it is done in a timely manner and accompanied with practical information that can help with remediation. In this paper we investigate ways to detect students at risk of failing early in the semester for timely intervention. The context of our study is a first year computer programming course. We explore whether the use of several student data sources can improve the process: submission steps and outcomes in an automatic marking system that provides instant feedback, student activity in the discussion forum Piazza and assessment marks during the semester. We built a decision tree classifier that is able to predict whether students will pass or fail their final exam with an accuracy of 87% mid semester. The obtained rules are useful and actionable for teachers and students, and can be used to drive remediation.

Keywords
Student performance prediction; classification of failing and passing students; automatic grading system; discussion board; assessment and feedback.

1. INTRODUCTION
Computer programming is an essential skill for software engineers and computer scientists, and also an increasingly required skill for graduates of many other disciplines, such as science, medicine, economics and business. Key factors in how well a person will learn programming include regular practice, as well as quick and efficient correction of mistakes and misconceptions. This means that students must be provided with tools that allow them not only to practice their programming skills but also to receive timely and useful feedback, which can be challenging, especially for large introductory computer programming courses. Lack of regular practice and sufficient feedback, often leads to students becoming uninterested or disheartened, and giving up learning to program.

Innovative technology-enhanced teaching and learning tools can help to solve this problem. In our introductory programming courses, we use a combination of an automatic marking and instant feedback system (PASTA) and a sophisticated discussion board (Piazza). These tools not only provide a semi-independent platform for students to build and test their knowledge, but also the opportunity for useful data collection and analysis, that can be used to improve teaching and learning.

In this paper we describe how data collected from these two sources, together with data from assessment marks, can be used to identify students who are at risk of failing and need more careful guidance, early enough so that remediation is possible. To illustrate this we use data from a large first year programming course. Specifically, the goal of this study is three-fold:

(i) to investigate whether students at risk of failing can be identified early enough in the semester for timely intervention, using machine learning prediction methods and information from three different sources: automatic marking system (PASTA), discussion board (Piazza) and assessment marks;

(ii) to investigate whether the information from the automatic marking system and discussion board helps improve the predictive accuracy, in comparison to just using the assessment marks;

(iii) to investigate how useful and actionable the produced rules are for remediation.

2. DATA SOURCES
An important characteristic of our study is that it triangulates data from three different sources that contain information not only about student performance, but also about student activities. Each source offers useful perspectives on student learning: progression in code writing and diagnostic (PASTA), interaction and engagement (Piazza), student performance (assessment marks).

PASTA is an automatic marking and feedback system developed in our school. It allows students to submit their solution for an assessment task online, checks this solution against public and hidden tests set by the teacher and provides immediate feedback to the student about which public tests were passed and failed. Students can then correct their mistakes and resubmit until all these public tests are passed. Feedback about the hidden tests is released when marking is completed, along with manual feedback.

The use of PASTA has resulted in better student engagement and improved learning, because of the instant feedback and multiple submissions. The PASTA data contains, for each task and student, all sequences of assessment submissions, the tests that were passed and failed (and why), the time stamps and mark obtained.

Piazza (www.piazza.com) is a mix of discussion board and wiki, allowing students and teachers to post notes, ask and answer questions individually or collaboratively. It was developed with the aim to connect students and promote classroom engagement. The Piazza data contains, for each student, the number of questions asked, answered and viewed, and the time and content of the posts.

The third data source includes all assessment marks during the semester and the final exam mark and is described in Sec.4.

3. PREVIOUS WORK
Previous work on predicting failure rate of students has been performed, normally by predicting exam grade just before the exam. Kotsiantis et al. [1] predicted final exam performance based on assignment marks throughout the semester in a distance education environment. This prediction was performed only at the end of the semester, and the attributes used would not allow for a

mid-semester prediction. They achieved an accuracy of 79% in predicting the final exam grade using an ensemble classifier.

Romero et al. [2] predicted final student marks based on Moodle usage data - the number of: quizzes passed and failed, assignments done, messages sent and read on the discussion board, and also the time spent on the assignments, quizzes and discussion board. They measured the geometric mean of the accuracies per class, which is an appropriate measure for imbalanced datasets as theirs, and achieved 67% with decisions trees. More recently, in [3] the same group investigated predicting the student grade (pass or fail) based on the student participation in a discussion forum, achieving accuracy of 75% using data collected in the middle of the semester and 90% using data collected at the end of the semester

Similar to student failure rates, student dropout rates have been studied, using a variety of assessment and non-assessment attributes. Agnihotri and Ott [4] predicted the likelihood of students dropping out of university after their first semester based on data provided such as admission information, placement tests and financial information. They were able to predict the retention of students with recall of 73% and precision 54%. Lykourentzou et al. [5] predicted dropout rate of students in an e-learning course environment, using the learning management system's extensive logs. They use machine learning techniques to achieve a 75-85% accuracy in the early sections of the course, and 97-100% accuracy in the final sections.

In this paper we extend previous work on predicting the students at risk of failing by using data from an automatic marking system and an advanced discussion board, in addition to assessment marks, from a computer programming course. We show how to define useful attributes from each data source, investigate if the student traces on the automatic marking system and discussion board help to improve predictive accuracy, and analyse how useful the prediction rules are for driving remediation.

4. CONTEXT OF THE STUDY

The study was conducted in the context of a large first year computer programming course with 223 students.

4.1 Assessment components

The six assessment components are summarised in Table 1.

Table 1. Assessment components

Homeworks	10	Weekly	Marks, Piazza
Task 1	2	Week 4	Marks, PASTA, Piazza
Task 2	6	Week 6	Marks, PASTA, Piazza
Practical test	16	Week 7	Marks, Piazza
Assignment	16	Week 12	Marks, PASTA, Piazza
Exam	50	Exam period	Marks, Piazza

The weekly homeworks were due before the computer lab and included multiple choice questions mainly requiring reading and understanding code. Their goal was to prepare students for the lab. The two tasks and assignment were programming assessments, with increasing level of difficulty, submitted via PASTA. Students were provided with skeleton code and required to complete the missing parts. The practical test involved writing code to solve five tasks with increasing difficulty levels in front of the computer. The exam, conducted at the end of the semester, was paper-based and required mainly writing code for solving problems. All assessment components were individual except for the assignment, where students had the choice of working individually or in pairs; 57% of students worked individually and 43% worked in pairs.

4.2 Predicted Variable

We predict the exam grade based on the marks of the other assessment components during the semester and the student activities on PASTA and Piazza. The two grades are defined as F (exam mark below 50, N=76), notF (exam mark of 50 and above, N=147). We chose the exam grade as a performance index because the exam: (i) is the major and most comprehensive assessment component, (ii) is conducted under strict conditions which minimises cheating, (iii) is independent of the other assessment components. The exam mark is highly correlated with the final mark (r=0.937).

4.3 Attributes

Table 2 summarises the student attributes that we defined to characterise student performance and activity.

Table 2. Attributes extracted from the three data sources

I. Assessment marks
homework_mark, task1_mark, task2_mark, prac_quiz_mark, assignment _mark (numeric) - mark (%) awarded for each assessment component
w7_homework_mark (numeric) – same as *homework_mark*, but only counting homeworks submitted before the end of week 7
II. PASTA activity – submission history
Starting and finishing times for assessments
task_start, task_finish, assignment_start, assignment_finish (numeric) – the average number of days before the due date that a student will start or finish the tasks or assignment
early_task, early_assignment (nominal, *yes/no*) - *yes* if the student starts the tasks faster than the average user; *no* otherwise
Multiple assignment submissions – improvement and consistency
marks_per_attempt_tasks, marks_per_attempt_assignments (numeric) – the average number of marks per PASTA submission of a task or assignment (including non-compiling submissions)
assignment_first_mark (numeric) - mark awarded for the student's first submission for the assignment
assignment_improvement (numeric) – the slope of the trendline of the student's assignment marks over each compiling submission; a larger number indicates rapid improvement
assignment_only_improvement (nominal, *yes/no*) - *yes* if the student's marks for compiling assignment submissions never decrease; *no* otherwise
assignment_consistency (nominal, multiple values) - goodness of fit (R^2) over each of the student's compiling submissions for the assignment, [-1, 1]; close to 1/-1 - linear increase/decrease in marks over submissions, close to 0 - random distribution of marks. Discretised as: *single* for single compiling submission, *none* for no assignment submission; *small/medium/high/very_ high* otherwise.
Pair work
pair_assignment (nominal, *yes/no*) - *yes* if the student worked in a pair for the assignment; *no* otherwise
Assignment submission statistics
single_submission (nominal, *yes/no/none*) - *yes* for one compiling assign. submission, *no* for more than one, *none* for no submission
assignment_total_submissions - total number of assignment submissions

assignment_compiling_submissions - number of compiling assignment submissions
III. Piazza activity – views, questions and answers
piazza_views, piazza_questions, piazza_answers (numeric) - number of posts viewed, questions asked or answered by the student on Piazza
piazza_activity (numeric) calculated as: (*piazza_views* + 10*(*piazza_ questions + piazza_answers)* + 5*(*piazza_posts - piazza_answers*)) / *total_posts*, where *piazza_posts* is the total number of contributions made by the student (asking or answering a question, or posting a comment), and *total_posts* is the total number of question threads on Piazza
piazza_active_viewer, piazza_active_questioner, piazza_active_answerer (nominal, *yes/no*) - *yes* if the student has an average or higher number of posts viewed, questions asked or questions answered; *no* otherwise
w7_piazza_views, w7_piazza_questions, w7_piazza_answers, w7_piazza_activity (numeric) and *w7_piazza_active_viewer, w7_piazza_active_questioner, w7_piazza_active_answerer* (nominal, *yes/no*) - same as the respective attributes without prefix *w7*, but only counting Piazza's posts up until the end of week 7

5. CAN WE PREDICT FAILING AND PASSING STUDENTS MID-SEMESTER?

We investigate whether we can predict accurately the students who will fail and pass the exam, based on the information available at two time points during the semester (and before the exam): in the middle of the semester (end of week 7) and at the end of the semester, just before the final exam (end of week 15). By the end of week 7, the students would have completed half of the homeworks, the two tasks and the practical test.

We built a Decision Tree (DT) classifier. One example in the data corresponds to one student and is described with the extracted attributes. An advantage of DTs is that the set of if-then rules they generate provides an explanation about the prediction which can be easily understood by teachers and students and directly applied.

Selecting appropriate attributes is very important for successful classification. Starting with the full set of attributes from Table 2, we used several methods for attribute subset selection [6] (manual and automatic such as correlation-based and wrapper, and combinations of them), before applying the DT algorithm. Although DTs have an inbuilt mechanism for attribute selection (only a subset of the attributes appear in the tree), their performance benefits from prior attribute subset selection. We report the best results. In all experiments, we used 10-fold stratified cross validation as an evaluation procedure.

Table 3 shows the accuracy results using data from all three sources and Figure 1 shows the generated DTs. The numbers in the brackets next to a leaf node in the trees give information about the coverage and correctness of the rule, e.g. (51/3) means that the rule covered 51 examples from the data, 3 of them we classified incorrectly and the remaining 48 were classified correctly.

Our results show that it is possible to predict the failing and passing students mid semester equally well as at the end of the semester – the two trees achieved the same accuracy, 87%. This accuracy is high enough to be useful in practical applications.

An examination of the confusion matrix shows that for the mid-semester tree the misclassifications are due to more failing students being classified as non-failing than the opposite. For the end of semester tree, there is no dominant misclassification type.

Table 3. Accuracy and number of rules using all three sources

	Marks + PASTA + Piazza
Mid sem. (week 7)	87.00 (8 rules)
End sem. (week 15)	87.00 (9 rules)

Figure 1 shows the two trees. Although equally accurate, the two DTs are different: they have different rules, using common and different attributes from the three sources. Both use *prac_quiz_mark* from assessment marks and *early_task* from PASTA but the other attributes are different, as shown below.

Mid semester (week 7)

```
prac_quiz_mark <= 81.875
|  prac_quiz_mark <= 54.375: F (51/3)
|  prac_quiz_mark > 54.375
|  |  w7_piazza_active_viewer = no
|  |  |  w7_homework_mark <= 70
|  |  |  |  early_task = no
|  |  |  |  |  task2_mark <= 70: notF (3)
|  |  |  |  |  task2_mark > 70: F (4/1)
|  |  |  |  early_task = yes: notF (2)
|  |  |  w7_homework_mark > 70: F (13/1)
|  |  w7_piazza_active_viewer = yes
|  |  |  task_finish <= 0: F (4)
|  |  |  task_finish > 0: notF (32/6)
prac_quiz_mark > 81.875: notF (114/3)
```

End of semester (week 15)

```
prac_quiz_mark <= 81.875
|  assignment_total_submissions <= 15
|  |  prac_quiz_mark <= 45: F (32)
|  |  prac_quiz_mark > 45
|  |  |  early_assignment = no
|  |  |  |  piazza_active_questioner = no
|  |  |  |  |  early_task = no: F (28/7)
|  |  |  |  |  early_task = yes: notF (3)
|  |  |  |  piazza_active_questioner = yes
|  |  |  |  |  assignment_finish <= 0: F (14/4)
|  |  |  |  |  assignment_finish > 0: notF (9/1)
|  |  |  early_assignment = yes: F (8/1)
|  assignment_total_submissions > 15
|  |  prac_quiz_mark <= 50.9375: F (2)
|  |  prac_quiz_mark > 50.9375: notF (13.0)
prac_quiz_mark > 81.875: notF (114.0/3.0)
```

Figure 1. DTs produced using all three data sources

The most important attribute in both cases is *prac_quiz_mark*, which is selected as a root of both trees and classifies correctly a large number of examples (e.g. If *prac_quiz_mark* > 81.875, then notF (114/3) in both DTs). This is expected as the practical quiz tests both theoretical and practical skills, and, similarly to the final exam, is conducted in a supervised environment, within time limits (in this case directly at the computer).

We highlight some interesting rules using attributes from PASTA and Piazza. From the mid-semester tree, the following rule shows the importance of following the discussions on Piazza, in addition to having relatively good marks on the practical quiz and homeworks:

If prac_quiz_mark ∈ (54.375, 81.875] &
w7_piazza_ active_viewer = no &
w7_homework_mark > 70
then F (13/1)

The following rule, also from the mid-semester tree, shows the importance of viewing the posts on Piazza, and also finishing the tasks earlier than on the due day, in addition to having a relatively good mark on the practical quiz:

If prac_quiz_mark ∈ (54.375, 81.875] &
w7_piazza_active_viewer = yes &
task_finish > 0
then notF (32/6)

From the end-of-semester tree, the following rule shows the importance of submitting the assignment and tasks early and asking questions on Piazza:

If prac_quiz_mark ∈ (45, 81.875] &
assignment_ total_submissions <= 15 &
early_assignment = no &
piazza_active_questioner = no & early_task = no
then F (28/7)

The rules in the two DTs generally make sense. The counter-intuitive ones (e.g. the two rules in the mid-semester tree that include *task2_mark* as their last condition and predicting F if *task2_mark* is greater than 70 and vice-versa) cover a very small number of instances (5/223 in this case) and represent coincidences in data rather than patterns.

Finally, both the mid-semester and end-of-semester trees are small (8 and 9 rules respectively), therefore easy to use by teachers.

In summary, the produced rules are compact, useful and actionable. They show the importance of the practical quiz, good practice such as starting and finishing assessments early and regularly reading the posts on the discussion board.

6. IS THE INFORMATION FROM PASTA AND PIAZZA USEFUL FOR PREDICTION?

We investigate if the information from the automatic marking system PASTA and the discussion board Piazza helps to improve the predictive accuracy, in comparison to just using the assessment marks. Table 4 shows the results when using marks only, and marks and PASTA only. The results using all three sources - marks, PASTA and Piazza - are given in Table 3.

Table 4. Accuracy and number of rules using assessment marks alone, and assessment marks and PASTA

	Marks	Marks + PASTA
Mid sem. (week 7)	84.30 (8 rules)	84.70 (13 rules)
End sem. (week 15)	82.96 (9 rules)	83.41 (14 rules)

We can see that using the assessment marks only provides a very good accuracy of 83-84%. The addition of information from the automatic grading system PASTA improves the accuracy by about 1%. Adding the information from the discussion board Piazza (Table 3) further improves the accuracy by about 3%, raising it to 87%. Hence, using information from PASTA and Piazza improves the predictive accuracy, in comparison to just using the assessment marks. However, this improvement is small in this case as the marks alone already provide high accuracy and there is a ceiling effect.

7. CONCLUSIONS

In this paper we investigate whether students at risk of failing can be identified early enough in the semester for timely intervention, using machine learning prediction methods and information from three different sources: automatic marking system, discussion board and assessment marks. We define useful attributes from each data source, to characterise student performance and activity. Using these attributes, we built a decision tree that achieved 87% accuracy in predicting whether students will pass or fail their final exam, from information available in the middle of the semester.

The produced rules are useful and actionable, and indicate the importance of starting and finishing assessments early and reading the posts on the discussion board, in addition to performing well on key assessment components. We show that using information from the automatic marking system and discussion board improves accuracy, compared to only using the assessment marks.

Our results can be used to detect students at risk of failing early in the semester and provide them with simple preventive feedback about remedial actions. Having an early flagging of students at risk also allows teachers of large classes to approach these students and provide more personalised remedial actions. At the beginning of the semester all students can also be made aware of the characteristics of the failing and passing students, to encourage better learning, good practice and improved student engagement.

An important aspect of our work is that we exploited different data sources capturing various facets of student activity during the course. This allowed the DT results to provide some concrete suggestions of remedial actions. The methodology we have followed can be applied to other contexts combining similar types of data sources. We are currently applying it to another very large course.

8. ACKNOWLEDGMENTS

This work was partially supported by the Human Centered Technology cluster at the University of Sydney.

9. REFERENCES

[1] Kotsiantis, S., Patriarcheas, K., and Xenos, M., 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems 23*, 6, 529-535.

[2] Romero, C., Ventura, S., Espejo, P.G., and Hervás, C., 2008. Data Mining Algorithms to Classify Students. In *Proc. Int. Conference on Educational Data Mining (EDM)*, 8-17.

[3] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S., 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education 68*, 458-472.

[4] Agnihotri, L., Ott, A., 2014: Building a student at-risk model: an end-to-end perspective. In *Proc. Int. Conference on Educational Data Mining Conference (EDM)*, 209-212.

[5] Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V., 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education 53*, 3, 950-965.

[6] Witten, I.H., Frank, E., Hall, M.A., 2011. *Data Mining - Practical Machine Learning Tools and Techniqu*e. Morgan Kaufmann.

Intelligent Tutor Recommender System for On-Line Educational Environments

Marian Cristian Mihăescu
University of Craiova
Department of Computer Science
Craiova, Romania
mihaescu@software.ucv.ro

Paul Ştefan Popescu
University of Craiova
Department of Computer Science
Craiova, Romania
sppopescu@gmail.com

Costel Ionaşcu
University of Craiova
Faculty of Economics and Business Administration
Craiova, Romania
icostelm@yahoo.com

ABSTRACT

This paper presents a method of using a classification procedure for retrieval of the most appropriate tutors in online educational environments. The main goal is assisting learners to find the most suitable colleagues that can provide them help. The retrieval is based on a user model built from experiences of previous generations of students. Performed activities represent the raw input data(i.e., the experiences), that one obtained from on-line educational environment. The goal of the developed system is to provide a list of colleagues that are willing and able to provide help. The student that is looking for a tutor will be aware of his weakness, his place among his colleagues and get some intuition regarding needed future activities that may improve effectively his knowledge level. The data processing for the retrieval mechanism is based on a classical classification engine that is custom designed for fulfilling the presented goal.

Keywords

Decision Tree, Classification Induction, Recommender System, e-Learning

1. INTRODUCTION

This paper addresses the problem of improving the knowledge level of a student that uses an online educational environment by using algorithms for indexing and retrieval mechanism. The proposed approach contains two main modules: the server side module where the indexing model is created and the client side where visualization and retrieval of a set of learners(i.e. prospective tutors) takes place. This set of colleagues represent the most appropriate options according to several predefined criteria specified by retrieval mechanism.

The first prerequisite for building a reliable recommender system is gathering high quality data in order to properly train the classifier, as main data processing unit within indexing mechanism. The activity related assets (e.g., database, log files, etc.) of the e-Learning environment are queried in order to obtain the training dataset. The assets must provide enough information such that all needed features that define students are computed and stored into the training dataset.

Once the input data for analysis is available the aim is to design a machine learning based recommender system that trains a classifier which acts as a core processing unit for the indexing mechanism. The next steps are choosing: the appropriate algorithm type (e.g., supervised, unsupervised, rule based, etc.), the algorithm itself, the features (e.g., name, meaning, type, values, etc.) and the overall setup necessary for obtaining a solution. In our case, we use supervised learning algorithms (e.g., classifier) and more exactly, decision trees [6]. The algorithm is used to classify new items, which in our educational context are represented by learners. The research issue of this paper regards designing and implementing a tutor recommender system. Addressing of this issue is accomplished by two means: properly designing a custom data analysis pipeline and building a tool that retrieves tutors in the practical context of Tesys [2] e-Learning platform.

For prototyping a general purpose classifier is used, i.e. a decision tree that is implemented in Weka [5] and this implementation is used for experiments. The key issues that are addressed regard properly setting up the general purpose classifier in a context of a practical problem in e-Learning application domain. Main ingredients for concept proof description and tool prototyping are presented in this paper along with detailed description of choices and their expected, observed and validated impact.

Once the recommender system is built, it can be used to obtain tutors for current or new learners by providing input only their computed values for the chosen features. On the client side, the learner is able to log in the application and access the tutor search utility application. The student first sees his class label in the existing model (i.e., his actual class) and then his target class which gathers the best suited tutors for him.

2. RELATED WORKS

The paper folds at boundaries between domains of machine learning and information retrieval as part of EDM and recommender systems. Educational Data Mining is an emerg-

ing discipline, concerned with developing methods for improving the relationship and interaction level between learners and professors. Since 2005 when the workshop referred to as 'Educational Data Mining' AAAI'05-EDM took place in Pittsburg, USA [4], followed by several related work-shops and the establishment of an annual international conference first held in 2008 in Montreal [1] many work has been done in this area [12] [11] .

Building recommender systems for e-Learning gathered many research efforts due to large number of practical usages within this application domain. Since e-Learning is a highly interaction domain, it is very appropriate for using Intelligent Data Analysis techniques for building various types of recommender systems. E-Learning personalization [8] represents one of the most common and general issues in e-Learning. Within this area of research issues like adapting the presentation and navigation [3], smart recommender in e-Learning [14] and various other commercial systems [9] proposes different input data, user modelling strategies or prediction techniques for reaching various business goals. Among the most used data analysis techniques there are content-based or item-based filtering, collaborative filtering, rule-based filtering, etc [10]. The general machine learning strategy of learning and predicting, information retrieval strategy of indexing and retrieval become in recommender systems for e-learning modelling and recommendation. Modelling regards thus users, content(i.e. questions, chapters, etc) and recommendation implies the existence of a implicit or explicit query. Once all these ingredients are put together in a consistent data analysis pipeline the output takes the form of a single recommended set [13].

Regarding involved technologies this paper uses Weka (Waikato Environment for Knowledge Analysis) as a popular suite of machine learning, data mining and information retrieval algorithms written in Java. The implemented algorithms are very flexible and can be used into the analyzing process of different kinds of data(from different domains). From Weka we have used J48 which is the implementation of the C4.5 [7] algorithm in Weka, a data analysis algorithm which generates a decision tree in order to classify data.

3. FRAMEWORK FOR INTELLIGENT TUTOR RETRIEVAL

The recommender module has the task of matching the query of a learner for a tutor against the existing data model. From software perspective the recommender module is a client for model builder module. Its main task is to produce results in such a way they may be intuitively displayed by the thin client application used by learner in his attempt to find a suitable tutor. Therefore the learner will see a tree-like structure due to native shape of the decision trees with actual class of the learner marked in red and with target classes marked in shades of green. In fact the green classes represent set of learners that are suitable for being tutors for learner that is querying the system.

In Fig.1 presents how the tutor recommender mechanism is designed as a data workflow. From interaction point of view students have to query the system that is integrated within Tesys-Web and after performing necessary operations on the server side they obtain the decision tree filled with

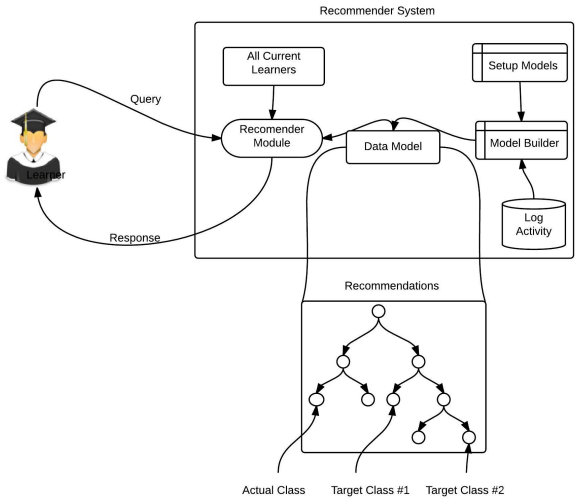


Figure 1: Activity use case diagram

prospective tutors. On the server side we have the business logic of the recommender system. Here the training dataset is built, thereafter the data model and the output as an xml file that can be displayed on the client side.

3.1 Description of Tasks within Recommender System

The main tasks performed within the recommender system regard preliminary offline data model building, setup of the recommender system, indexing currently existing learners into already created model and computing and extracting relevant tutors by applying the already setup recommendation strategy.

3.1.1 Learners Modelling Phase

We apply machine learning (i.e., decision trees) techniques to build learners profiles by using already existing implicit performed activities from usage sessions of learners that used the system in previous years. This data represents the training data and the output is represented by a set of classes (i.e. leaves in the decision tree) such that each class corresponds to a learners profile. Once sessions and corresponding activity data are delimited in such a way that all features describing a learner are processed, we can use the decision tree builder to obtain the baseline data model. Once the data model is created the currently existing learners within the e-Learning platform are placed in their corresponding leaves. At this moment currently existing learners are indexed in the decision tree data structure and are ready to be queried.

3.1.2 Tutors Recommendation Phase

The query of a learner for a tutor is regarded as a parametrized implicit query. The parameters aim tuning the retrieval mechanism such that optimal solutions are returned from solution space. The solution space is regarded as a set of classes(i.e., leaves) each class containing a set of prospective tutors. The set of classes need to fulfill one basic requirement, which is to be labeled with a "better" class label that

the one in which the querying learner resides. A total order set of classes is ergonomically computed with the first item containing learners "close" to the querying learner and the last item containing learners "further" to the querying learner. In this context parameter tuning will manage to decide a certain number of prospective tutors that are picked up from one of the target classes. Intuitively, choosing tutors from a class that is "closer" to the querying learners' class will return tutors with a profile that is better but similar. On the other hand, choosing tutors from a class that is "further" to the querying class will return tutors with the best profiles among all colleagues.

3.2 Description of the Data Analysis Process

The main concept considered in the model is the "Learner", which is also a "Tutor". Once the data model is built from the training data the current set of learners $L=\{L_1,L_2,L_3...L_n\}$ is distributed in corresponding classes according with the key feature values $f_{i,k}$. For current prototype implementation the features are not weighted since the decision tree itself provides a ranking in feature selection.

All classes of learners are considered as resources for which an *"affinity"* function needs to be defined in order to retrieve the most suitable tutors. Defining the *affinity* function needs to take into consideration several criteria such a better overall knowledge weight, specific values in communication related features(i.e., messaging activity, forum activity, etc) and demographic features. Due to its specific topology of the decision tree also ranks the leaves in classes. A normal distribution function is defined such that the lowest ranked class is assigned 0 knowledge weight and highest ranked class is assigned a value of 1 knowledge weight. All in between classes get a knowledge weight ranking between 0 an 1.

Thus the data analysis task is to identify the actual class of the learner who is querying for a tutor and to provide most suitable options from the subsequent classes in obtained ranking of the current learners. With this approach the tutor retrieval becomes a matter of properly specifying querying parameters. The proposed mechanism offers the possibility of obtaining any of the feasible solutions, somewhere between the very next learner which resides in the class with the next knowledge weight value up to the top class learners in ranking. From this perspective several parameters are defined. One manages the proximity of the class from which tutors are retrieved.

4. EXPERIMENTAL RESULTS

The main input of the server application is the activity repository. This raw data taken from the database is converted to an *.arff* file, which is used to train the classifier.

In Fig. 2 are presents the meaning of the attributes from the *.arff* file and Fig.3 presents the obtained decision tree based on the training dataset. Several functionalities were developed in order to be able to load and parse the decision trees. One of them is successor computation. Because of our dataset structure, the decision tree contains ordered leaves. This functionality is successfully used for fulfilling specific user constraints. For example, if the student does not want

userid	The identification number of the student
nrHours	The total number of hours spent on the platform
avgMark	The average mark obtained at a discipline
messagingActivity	This feature represents a discretization of the on-line involvement regarding sent and received messages. A messaging activity greater than 50% of the average number of sent and received messages means YES and smaller means NO.
noOfTests	The total number of tests taken on the platform
avgMessageLength	This feature represents a discretization of the average message length. An average length greater than 160 characters means the student sends LONG messages and smaller means SHORT messages. This value has been chosen with respect to the limitation of the SMS message.
class	The class that the user belongs to, which can be low or high. Low/high values were assigned based on the final result obtained by the student.

Figure 2: Features

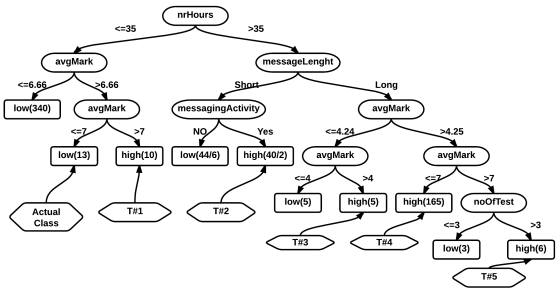


Figure 3: Tree example

to have a step by step progress, he will be able to use this feature to retrieve tutors which are i steps ahead of him.

Here is the pseudocode for the method used to locate the classified student's actual and target class. This recursive method takes two parameters: a node element containing information from the xml and a boolean variable, stating whether the parent has been marked or not. A node is marked when the student meets the requirement stated inside the node (for example if the "messageLenght" is "LONG").

```
function studentSearch(Node parent, boolean isMarked)
{
   SET nodeList to the list of child nodes of "parent"
   FOR each node in nodeList
      IF is element node THEN
         SET atr to the value of the attribute "attribute"
         IF atr is null THEN
            IF isMarked THEN
               add new atrribute to the node
            END IF
            IF "decision" attribute is "high" THEN
               SET target to the current node
            END IF
         ELSE
            SET expresie to ""
            SET belongs to false
            CASE atr IS
               "avgMark":
                  generate the expression to be evaluated
                  SET belongs to the result of the evaluation
               "nrHours":
                  generate the expression to be evaluated
```

```
                SET belongs to the result of the evaluation
               ..............................................
            END CASE
            IF belongs AND isMarked THEN
               add new atrribute to the node
               CALL studentSearch WITH current node, true
            ELSE
               CALL studentSearch WITH current node, false
            END IS
         END IF
      ELSE
         REMOVE node FROM nodeList
      END IF
   END FOR
}
```

The thin client automatically computes the class of the student who is searching for a tutor. These values are used to determine the actual class of the student. The actual class of the student is marked with red and the target class is marked with green.

In Fig. 4, the inspection of the green node reveals to the student a list on colleagues that may help him as tutors. The student has a messaging system to his disposal for contacting his recommended tutors in an attempt to find answers from the right persons.

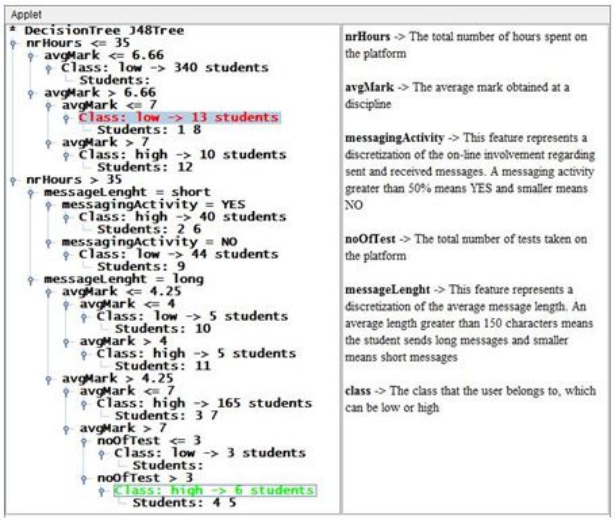


Figure 4: Prototype of the tutor retrieval system

Let us consider student S1, which has already used the e-learning platform and there is enough data logged about him, including the following values for the attributes: nrHours = 30, avgMark = 6.80, messagingActivity = No, noOfTests = 2, avgMessageLength = SHORT. After running the tool, he finds out that his actual leaf/class is the second one when parsing the tree from left to right, so he has to put some effort to catch up with his colleagues. Assuming he wants to spend the necessary time to gradually learn from his colleagues, the system can recommend him tutors belonging to the third leaf of the tree (1^{st} level successors). He will therefore receive the contact details of the students from that leaf. After managing to improve his performances and become himself part of that leaf, he will be able to continue to the next step. If he however wants to try to learn from the best directly, the system will be able to provide him with the contacts of the best tutors available (belonging to the green leaf in Fig. 4).

5. CONCLUSIONS

The paper presents a custom approach for providing assistance to learners in on-line educational environments. The assistance regards the option of finding colleagues that may offer guidance in respect to activities that need to be performed for improving the student's knowledge level.

Each student will benefit by using this tool from the perspective of improving their knowledge. It will be easier for students to communicate with someone their own age, ask questions about the things they don't understand, and get more clarification and feedback.

6. REFERENCES

[1] T. B. J. Baker R.S.J.d.; Barnes. Educational data mining 2008 :1st international conference on educational data mining proceedings. *Educational Data Mining*, 1:20–21, 2008.

[2] M. M. C. Software architectures of applications used for enhancing on-line educational environments. *Manuscript submitted for publication.*

[3] H. Chorfi and M. Jemni. Perso : Towards an adaptative e-learning system. *Journal of Interactive Learning Research*, 15(4):433–447, 2004.

[4] B. J. Proceedings of aaai2005 workshop on educational data mining. 2005.

[5] W. I. F. E. T. L. H. M. H. G. C. S. Jo. Weka: Practical machine learning tools and techniques with java implementations. *University of Waikato, Department of Computer Science*, 1999.

[6] J.R.Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[7] J.R.Quinlan. C4.5: Programs for machine learning. *San Mateo, CA*, 1993.

[8] M. Khribi M. K.; Jemni and O. Nasraoui. Recommender system for predicting student performance. *Educational Technology & Society*, 12(4):30–42, 2009.

[9] B. Mobasher. Data mining for web personalization. *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer-Verlag, Berlin-Heidelberg, 2006.

[10] O. Nasraoui. World wide web personalization. *Encyclopedia of Data Mining and Data Warehousing*, 2005.

[11] C. Romero and S. Ventura. Data mining in e-learning (advances in management information). *WIT Pr Computational Mechanics*, 2006.

[12] B. R. S. and Y. K. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining (JEDM)*, 1:3–17, 2009.

[13] O. N. Z. Z. E. Saka. Web recommender system implementations in multiple flavors: Fast and (care) free for all. *Proceedings of the ACM-SIGIR Open Source Information Retrieval*, 2006.

[14] O. Zaiane. Building a recommender agent for e-learning systems. *Proc. of the 7th International Conference on Computers in Education,*, 3-6:55–59, 2002.

Discovering the Pedagogical Resources that Assist Students in Answering Questions Correctly – A Machine Learning Approach

Giora Alexandron
Massachusetts Institute of Technology
giora@mit.edu

Qian Zhou
Tsinghua University
zhouqian@gmail.com

David Pritchard
Massachusetts Institute of Technology
dpritch@mit.edu

ABSTRACT

This paper describes preliminary results from a study in which we apply machine learning (ML) algorithms to the data from the introductory physics MOOC 8.MReV to discover which of the instructional resources are most beneficial for students. First, we mine the logs to build a dataset representing, for each question, the resources seen prior to each answer to this question; Second, we apply Support Vector Machines (SVMs) to these datasets to identify questions on which the resources were particularly helpful. Then, we use logistic regression to identify these resources and quantify their *assistance value*, defined as the increase in the odds of answering this question correctly after seeing the resource. The assistance value can be used to recommend resources to students that will help them learn more quickly. In addition, knowing the assistance value of the resources can guide efforts to improve these resources. Furthermore the order of presentation of the various topics can be optimized by first presenting those whose resources help on later topics. Thus, the contribution of this work is in two directions. The first is Personalized and Adaptive Learning, and the second is Pedagogical Design.

Keywords

Adaptive Learning, Pedagogical Design Optimization, MOOCs

1. INTRODUCTION

A central question in online courses, as in education in general, is how to design measurably more effective pedagogy. Since online courses, and specifically MOOCs, offer "full course" environments and produce log files that can be analyzed by computational tools, it is only natural that such tools would be used to optimize online pedagogy. While most pedagogic design in online education is based on 'best practices' and subjective opinions, e.g. [4], we concur with Koedinger et al. [3] that optimizing the design of instructional resources is an area in which ML and educational data mining (EDM) techniques can add a lot of value.

We propose a machine-learning, data-driven method that yields various kinds of analytics that can be used by course designers to improve their courses. Specifically, our work concentrates on computing the assistance value of instructional resources. Seaton et. al. [6] showed that the resources used for homework and exam problems differed dramatically, but did not evaluate the *effectiveness* of the selected resources. Our method aims at discovering exactly this – the contribution of particular instructional resources (e.g., page 121 in the e-text) for solving specific questions. From this, various other measures can be derived, such as which resources are generally useful, which questions do not have good supporting resources, etc.

Our longer term vision is that this can be used to augment educational resources with meta-data describing their contribution to various tasks, in line with Mccala's 'Ecological Approach' [5]. This approach suggests using ML and EDM to automatically infer the educational value of on-line resources in order to combine them to achieve educational goals. Inspired by this, Champaign and Cohen [1] presented an algorithm for sequencing educational resources based on their educational value for a specific knowledge unit. Our work suggests means for computing these values, which their algorithm takes as an input.

In the context of personalization, a lot of work has been done in predicting performance and sequencing questions, for example the interesting algorithm of Segal et al. [7]. Our preliminary results show that considering the particular educational resources that students used can also improve the prediction of their performance. This is especially relevant in MOOCs, since the students are free to choose their path through the course, and can attempt a question without going over the pedagogical resources that are important for solving it.

Our approach is based on a two-step method for computing the assistance value of instructional resources. The first step aims at identifying questions that have strong connection to the course resources. The strength of the connection between a question and its resources is operationalized as the difference between the accuracy of a prediction model that considers resources seen prior to attempting the question *and* previous performance, and the accuracy of a model that considers *only* previous performance. On such questions we conduct a second step, aiming at identifying *which* are the contributing resources and quantifying their value. The results have two immediate payoffs. One is optimizing the course design. The other is content recommendation.

The rest of this paper is organized as follows. Section 2 describes our method in detail. Section 3 presents preliminary results obtained from running the method on the Introductory Physics MOOC 8.MRev. Section 4 discusses limitations, and Section 5 presents directions for future work.

2. OUR APPROACH

This section is organized as follows. First, we define the notion of assistance value and what we consider as resources. Second, we give a high-level description of the process for calculating the assistance values. Third, we describe in more details the steps – knowledge representation, data mining, and the ML algorithms.

2.1 Resources and Assistance values

The assistance value Rq is a measure of how much a particular pedagogical resource R (say, a video explaining gravity) contributes to solving question q. It is defined as the increase in the odds that a student seeing R will solve q correctly.

The resources considered in this study are either html pages containing textual explanations, instructional videos, or questions.

2.2 High-Level Description

The process for discovering the assistance values consists of the following steps:

I. Prepare a list of the pedagogical resources from the course structure files.

II. Mine the raw data (students' logs) to create a dataset representing the resources that the students interacted with before attempting the questions.

III. Identify questions in which the resources have a significant contribution to students' success. To achieve this, we compare, for each question, the predictive power of a SVM model that considers the resources seen before attempting this question to a baseline SVM model that considers only the aggregated performance on questions attempted before this question.

IV. For each question identified in step III, discover which resources have the highest assistance value. To achieve this, we use a logistic regression with the resources as independent variables and success/failure as the dependent variable. Then, the exponents of the coefficients are interpreted as the assistance value of each resource.

2.3 Data Mining and Knowledge Representation

As first step, we build, per question, a dataset representing the resources that the student interacted with before each attempt to each question. More specifically, we use a binary feature space, with each feature representing whether a resource was seen or not. Each attempt makes an example, with '1's for the resources seen before answering, and success/failure as the binary tag of this example. Since some of the questions allow multiple attempts, a student might contribute more than one answer to a question.

We note that we chose to start with the simplest representation, and operationalized 'interacting with a resource' as a two-state condition – seen or not. We deliberately decided to use a representation that does not preserve information such as the order in which the resources were seen, the amount of time spent on each resource, and other relevant aspects of the interaction between a student and a resource, as encoding them has an exponential effect on the size of the feature space.

Performance as an additional feature. Student's ability is an important factor when it comes to predicting performance. Thus, we add it as a feature to the model. Student's ability was operationalized as percentage of success on previous attempts.

Preparing the Data. The data mining algorithm, implemented in Python, works as follows*:* For each time-sorted student log file, the algorithm scans the log while maintaining, per student, a list of the resources seen so far and an ability parameter. Each time a resource is accessed, it is added to the list (unless it is already there). Each time a question is attempted, the algorithm adds to the dataset of this question a new vector with the resources seen, the ability parameter, and a tag indicating whether this attempt was successful or not. Then the algorithm updates the ability parameter.

Exploring Various Models. To achieve the best results, we consider various models, which differ on the 'length of their memory', namely, how many resources they keep in the list. For example, a model with memory_length=5 considers only the last 5 resources seen before each attempt. Thus, for each question we actually build several datasets, one per memory_length value. The values that are considered are 1/2/3/5/10/1000. A dataset with memory_length=0 is also prepared, for benchmarking (see next section). This dataset does not 'remember' resources, only student's aggregated performance (student's ability) before attempting the question. We denote the dataset of length *j* for question *q* with *Dqj* (and omit *q* when referring to this dataset for all the questions).

The rationale underlying testing various options is mainly that we assume that some questions might require many resources, while for others, a 'long memory' might include a lot of irrelevant data.

2.4 Using SVM as a Filtering Scheme

To find questions for which the instructional resources used are significant, we train and test (using a standard 10-fold cross-validation) for each question *q* a SVM model on each of the datasets *Dqj*, for *j* = 0/1/2/3/5/10/1000 (we denote the SVM model trained on dataset *j* of question *q* with *Mqj*, and omit *q* in case we refer to this model in general). The baseline described in the previous subsection is *Mq0*. Model accuracy is measured as the average accuracy of the 10-fold cross-validation and denoted *accuracy(M)*. We then compute the relative improvement that each of the models *Mqj*, *j* = 1/2/3/5/10/1000, give over Mq0, and pick the model that gives the highest *relative improvement*, defined as $\frac{accuracy(Mj)-accuracy(M0)}{1-accuracy(M0)}$. We consider questions on which the best model gives more than 10% relative improvement as questions with strong connection to the course resources.

2.5 Using Logistic Regression to Compute Assistance Values

As described above, the role of the Logistic Regression is to identify the resources with highest assistance value for each question. This step is conducted as follows. For each problem found by the SVM to have a strong connection with the resources, we train a logistic regression on the dataset that produces the best SVM model. For example, if for a specific question *q* the most accurate model was *Mqj*, we train a logistic regression on *Dqj* (in case several SVM models give the same performance, we follow Ockham's Razor rule and take the lowest j).

The result is that per question *q*, we have a logistic model that predicts the probability of answering *q* correctly as a function of the resources seen and the ability. As described above, the coefficient attached to each feature quantifies the contribution of this feature to the final outcome, with the *p value* representing the level of confidence.

The coefficient attached to each feature is interpreted as the *assistance value* of the resource that this feature represents, and we consider only those with high confidence (defined as *p value* < 0.05).

We note that an alternative approach was to use one method both for the prediction and for quantifying the value of the resources. This approach was tried with logistic regression and with Decision Trees, which are easily interpretable machine-learning methods. However, the prediction accuracy gained by these methods was relatively low, comparing to the accuracy achieved by SVM, which on the other hand, is a much less interpretable model. Thus, we separate the process into two phases, one aims at prediction and built on SVM, and one aims at quantifying the assistance values and built on logistic regression.

In Section 5 we discuss various ways in which the prediction models and the assistance values can be used for pedagogic design and recommendation.

3. CASE STUDY – INTRODUCTORY PHYSICS MOOC 8.MReV

Context. We applied the above method on the data obtained from the 2014 instance of the introductory physics MOOC 8.MReV given by the third author and his team through the edX platform. The course attracted about 13500 students. Gender distribution was 83% males, 17% females. Education distribution was 37.7% secondary or less, 34.5% College Degree, and 24.9% Advanced Degree. Geographic distribution includes the US (27% of participants), India (18%), UK (3.6%), Brazil (2.8%), and others (total of 152 countries). (All numbers are based on self-reports.) The course lasted for 14 weeks, with content divided between 12 mandatory units and two optional ones. From the course structure file we extracted 1362 pedagogic resources (1020 problems, 273 pages, 69 videos).

Data Mining. We considered 1308 questions for which there were more than 100 student attempts. (For problems that contain several graded sections, we consider each of them as a question. Thus this number is bigger than that in the previous paragraph.) We used the logs of all the students who attempted these questions rather than restricting to those students who exceeded a particular benchmark of participation. As described in Subsection 2.3, for each question we created 7 datasets, each representing a different 'memory length'.

SVMs and choosing the questions. On the next step, we trained a SVM model on each of the datasets, i = 0/1/2/3/5/10/1000 as described in Subsection 2.4, using *R*'s libsvm [2]. This yields seven SVM models for each question, each tagged with its accuracy level. The results show that in overall, models *M1…10* performed better than *M0*, which was always at least good as majority-class prediction. This was evaluated using a paired one-side t-test that tested the hypothesis that the accuracy of *Mi* over all questions is higher than the accuracy of *M0* on all the questions, for i=1,2,3,5,10. For *M1000*, the null hypothesis was not rejected, so we cannot say that in general this model behaved better than performance-based prediction. We believe that the main explanation for this is that considering resources used long before the question at hand was even opened introduces a lot of noise into the data, reducing the weight of the proximate resources. This is exemplified in Figure 1, which shows, for 5 typical questions, the relative improvement that models 'remembering' i=1,2,3,5,10,1000 previous resources give relative to remembering only aggregated performance of each student.

Next step was to choose, per question, the best model. Figure 2 shows, per question, the relative improvement of the best model compared with the accuracy of *M0* on this question. We took relative improvement > 10% as the cut-off for defining questions with strong relation to the pedagogic resources (the line is marked in the figure). In total, of 337 questions were above this threshold.

Logistic Regression. For each of the questions identified by the previous step, we trained a logistic regression on the data that produce the best SVM model, using standard packages in *R*. For example, if for question q the best SVM was *Mi*, we trained a logistic regression for q on *Dqi*. For each question, we sorted the coefficients with p *value* <0.05 in decreasing order. This yields the assistance values. Table 1 shows an example of the two most significant resources found for a question from homework 12, which deals with gravity and orbits. According to the model, the two most significant factors that correlate with answering this question correctly are seeing the html page Angular_Momentum_of_Orbits, which explains content related to this question, and student's performance on previous questions.

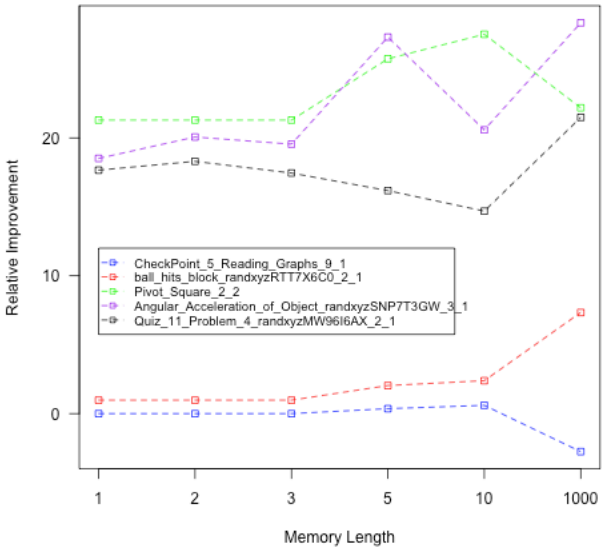


Figure 1. Relative improvement vs different memory length.

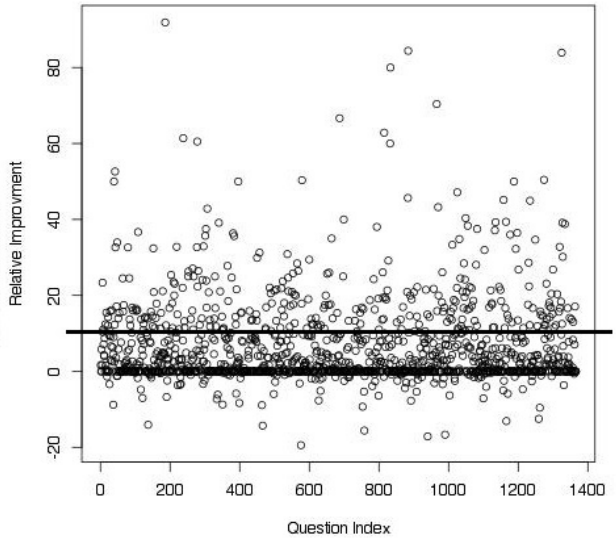


Figure 2. Relative improvement.

Table 1. Example of two most significant predictors

Question: : Homework 12, Gravity and Orbits				
Resource name	**Estimate**	**Std..Error**	**t.value**	**Pr...t..**
Angular_Momentum_of_Orbits	0.73	0.36	2.03	0.042
performance	0.62	0.2	3.05	0.003

Validation. In order to evaluate the meaningfulness of the results, we executed an expert validation protocol aimed at measuring the *precision* of the algorithm. In our case, *precision* is defined as the fraction of retrieved resources that are relevant. We gave to one of the course designers a list of 10 questions, each with 3-5 resources found to have assistance value. The course designer was asked to mark whether each resource is irrelevant/slightly-relevant/highly-relevant to the question. Weights were 0/0.5/1, respectively. In total, the precision on this sample, according to the expert, was

42.5%. We did not measure *recall*, which is the fraction of relevant resources that are retrieved, and is typically used in conjunction with *precision*, since the number of relevant resources for a specific question is unknown, and some of them can be interchangeable. Due to lack of space, we omit a detailed analysis that was done with the expert on the results given for a specific question.

4. LIMITATIONS OF THE MODEL

Our model has limitations in several areas. Due to lack of space we present them very briefly.

Cognitive. Currently the model makes simplistic assumptions on the nature of knowledge acquisition and retention. For example, it does not give any weight to the *time* spent on the resource, the *time since* seeing the resource (knowledge can be forgotten), order of resources is not considered, and it is assumed that the relation between the resources is additive (we used SVM with a linear kernel).

Model. Another limitation is that some of the independent variables in our model are collinear (i.e., A is a resource of B; A and B are resources of C). One effect on Logistic Regression is that the ability to infer the value of specific coefficients is reduced. A possible remedy is discussed in Subsection 5.2.

Data. As typically happens in real world examples, our data is skewed. For example, many of the participants already know the material (i.e., Physics teachers taking the course for professional development), so the resources they see have low effect on their ability. This adds a lot of noise to the data. Also, the ratio of examples-to-features is about 1:1, far from optimal.

5. FUTURE WORK

In this paper we described a method for computing the assistance value of pedagogic resources, presented preliminary results, and discussed limitations. Below we present directions for future work, which include further evaluation of the use of the various applications of this method, and removing limitations.

5.1 Using the Assistance Values

Finding the assistance value of resources will be useful for Pedagogical Design and for constructing Recommender engines.

5.1.1 Pedagogical Design Optimization

The assistance value can be used to address several interesting issues:

What types of resources are most effective: resources that have significant assistance value for a number of questions tell us what learning to emphasize. We can also determine the characteristics of resources that are most helpful - e.g. types (videos vs. e-text) or topics (momentum vs. energy).

Questions that lack good resources: If questions lack resources that help students to solve them this might indicate that the designer should add or improve (or possibly move closer to that question) the resources that ought to help.

Identifying redundant/bad instructional resources: If a particular resource is of little assistance for all questions, it is probably a distraction from good instruction (or covers a topic not assessed by any question).

Location of resources: Good resources that are located far from the question that they support may help students learn foundational skills.

5.1.2 Recommender Systems

In the future assistance values can be used for constructing an online resource-recommendation engine. Before a student attempts a question, the engine could use the logistic model to predict the probability that a student will get it correctly. In case this is low, a list of resources can be provided, recommended based on their assistance value, with simple metadata about each (e.g. whether e-text, a worked example, a video… as well as the median time students spent on it). This would allow the student to select the type of resource they prefer. Furthermore it would enable us to obtain much more data on the effective resources so we could determine which were best for students with different overall abilities and even possibly with different learning preferences.

5.2 Removing Limitations

Logistic regression is used both for its interpretability – to get the assistance values, and for its probabilistic classification – to predict the probability that a student will answer a question correctly. If this probability is low, we can recommend the resource with the highest assistance value that was not seen yet. One direction that we investigate is to separate this between two models – one for interpretability and another for probabilistic classification. This will allow considering other models, such as probabilistic SVMs. We note that for recommendation only, a strong probabilistic classifier is enough, and knowing the assistance values explicitly is not necessary. The process for finding the best recommendation is simple. For each unseen resource r, the engine will run the classifier on a vector consisting of the resources seen so far + r, and will recommend the resource that leads to the highest probability.

6. ACKNOWLEDGMENTS

This work is supported by a Google Faculty Award and MIT. We would like to thank Christopher Chudzicki, Zhongzhou Chen, Sait Gokalp, and Youn-Jeng Choi for useful comments, and to edX for accessing the data.

7. REFERENCES

[1] J. Champaign and R. Cohen. Ecological content sequencing: from simulated students to an effective user study, 2013.

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines, 2011.

[3] K. R. Koedinger, E. Brunskill, R. S. J. de Baker, E. A. McLaughlin, and J. C. Stamper. New potentials for data-driven intelligent tutoring system development and optimization, 2013.

[4] T. L. Leacock and J. C. Nesbit. A framework for evaluating the quality of multimedia learning resources, 2007.

[5] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners, 2004.

[6] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? 2014.

[7] A. Segal, Z. Katzir, K. Gal, G. Shani, and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning, 2014.

Using Topic Segmentation Models for the Automatic Organisation of MOOCs Resources

Ghada Alharbi
Department of Computer Science
Sheffield University
Sheffield, S1 4DP, UK
galharbi1@sheffield.ac.uk

Thomas Hain
Department of Computer Science
Sheffield University
Sheffield, S1 4DP, UK
t.hain@sheffield.ac.uk

ABSTRACT

As online courses such as MOOCs become increasingly popular, there has been a dramatic increase for the demand for methods to facilitate this type of organisation. While resources for new courses are often freely available, they are generally not suitably organised into easily manageable units. In this paper, we investigate how state-of-the-art topic segmentation models can be utilised to automatically transform unstructured text into coherent sections, which are suitable for MOOCs content browsing. The suitability of this method with regards to course organisation is confirmed through experiments with a lecture corpus, configured explicitly according to MOOCs settings. Experimental results demonstrate the reliability and scalability of this approach over various academic disciplines. The findings also show that the topic segmentation model which used discourse cues displayed the best results overall.

1. INTRODUCTION

In recent years, Massive Open Online Courses (MOOCs) have been in the spotlight of the media, education professionals, entrepreneurs and technologically aware members of society. As a result, leading universities have been convinced to run their courses online, by establishing open learning platforms, as seen with MIT Open Course Ware (OCW)[1] and Open Yale Courses (OYC)[2].

The majority of these learning platforms organise their resources in line with a pedagogical model, which will allow easy online browsing and accessing [23]. On the other hand, organising these resources takes a great amount of time and is platform dependent, and a large percentage of these platforms have varying formats and structures of the pedagogical model they are based on [23]. In order to decrease the above efforts, unstructured text can be automatically split into coherent sections, which are thus more suitable for online browsing. As these sections include the content of the learning units, an automatic pedagogical annotation model can be employed to organise these units into introductions, descriptions, explanations, examples and other pedagogically significant notions, as examined by [14]. Even though the use of automatic pedagogical annotation models appears suitable, a number of MOOCs sources are structured in line with both the pedagogical and topical approaches. An example of this would be Figure 1(a), the physics lecture from OYC, which displays both ways of structuring. The first and fifth sections depict the pedagogical elements, while the remainder includes the topic segments. This can also be seen in the economics lecture in Figure 1(b).

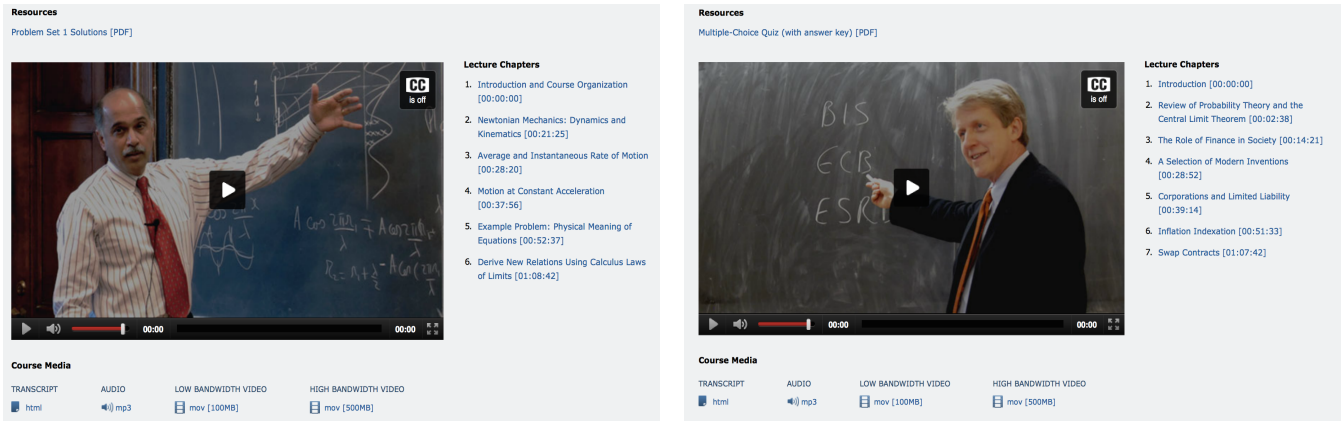
This paper will examine the use of state-of-the-art topic segmentation models to structure lecture resources into cohesive segments, making them suitable for MOOCs content browsing. To evaluate the segmenting applications in the proposed scenario, a test corpus was established using two different disciplines, which were physics and economics, derived from the OYC platform [25, 21]. The topic segmentation models employed in this research include similarity-based models, as seen in [3, 16, 11], language model-based, such as [7, 28] and topic model-based, as seen with [6, 22]. The key strengths of this methodology are its discipline and platform neutrality, which are highlighted in the results of this study. Furthermore, the impact of lexical and discourse cues were examined as features of the segmentation model.

It can be seen from the outcomes that the topic segmentation model which used discourse cues, together with lexical features, showed superior results for the two disciplines. This is due to the fact that discourse cues are often employed to signal the lecturer's aim of the discourse, which means that their learning units are represented more effectively [9]. Despite this, further analysis is required, since the current topic segmentation models hypothesise that discourse cues occur only at the start of an utterance, as seen in [18, 11, 7]. However, other studies have noted that discourse cues can occur at any point in an utterance, and they are a small part of a larger linguistic expression of a writer or speaker [27, 5].

2. BACKGROUND

A number of studies have shown how an automatic pedagogical annotation can be applied to organise lectures resources [14, 24]. However, instead of introducing new aspects such as pedagogical concepts, this paper examined the

[1]http://ocw.mit.edu/index.htm
[2]http://oyc.yale.edu



(a) (b)

Figure 1: Examples of the Interface used for browsing (a) physics and (b) economics lectures in OYC [21, 25].

wider applicability of topic segmentation models for structuring MOOCs content into cohesive units suitable for browsing. In turn, this section concentrates on the work of topic segmentation models, and specifically unsupervised topic segmentation, for either written or spoken language. There has been extensive research on unsupervised segmentation of text, based on lexical cohesion, but certain studies tried to involve other elements, such as discourse or visual cues [7, 8]. This paper will focus mostly on how lexical cohesion is modeled either as similarity-based, language model-based or topic model-based.

TextTiling [12] is considered the first similarity-based model to calculate the cosine similarity between two adjacent blocks of words based purely on word frequency. C99 [3] is based on divisive clustering with a matrix-ranking scheme, while LCSEG modeled lexical chain repetitions of a given lexical term, throughout a fixed-length window of sentences and then chose segmentation points at the local maxima of the cohesion function [11]. MCS [16] optimised normalised minimum-cut criteria, centred on a variation of the cosine similarity between sentences.

An early language model-based algorithm, UI, has been proposed by [28], who tried to find segmentations with compact language models. Furthermore, [7] employs a generative Bayesian model BSEG for topic segmentation. The algorithm computes the maximum likelihood estimates by looking at the entire sequence of sentences, at specific topic boundaries. Also, the model utilises the initial of the potential boundary utterances as discourse cues for the unsupervised model, which is an extension of the work by [11], who automatically identified discourse cues using true labeled boundaries in a supervised fashion.

Latent Dirichlet Allocation (LDA) is a generative model which uses latent structures to model the underlying similarities among observations and it is widely adopted in text analysis to model the shared topics among documents [2]. Topic model-based segmentation was initially interpreted by [26] and built upon by [17]. The most recent LDA based segmenter is TopicTiling [22], which undertakes linear topic segmentation with a pre-trained LDA topic model and estimates the similarity between segments to evaluate text coherence, based on a topic vector representation with cosine similarity. Only the most common topic ID is given to every word in a sentence through Gibbs sampling, in order to maintain efficiency. [6] have shown a hierarchical Bayesian model, which makes use of both Bayesian segmentation and structured topic modelling STM. Superior performance over various models, in both written and spoken texts [6], has been seen with this model. Likewise, the segmentation method of PLDA [20] samples segment boundaries, but also jointly samples a topic model.

The applications of topic segmentation models range from information retrieval to topic tracking [13], summarisation [14] and segmentation of multi-party conversations [11, 20].

3. METHODS

3.1 Data Preparation

Under the Creative-Common license, freely accessible lectures on the OYC website are used as data sources. Expert speakers conducted the lectures, and appear as high quality video and audio data, transcripts, subtitles and lecture segmentation on the course website, as part of MOOCs's initiative. Examples of this segmentation in physics and economics lectures are illustrated in Figure 1. High-level structure distinguishes the lecture as shown in the segmentation. These labelled segments boundaries used as the reference dataset to evaluate the models performance. From these data sources, the two distinct disciplines of physics and economics were selected to establish a new dataset. During the preparation of this study, the total sum of lectures was 47, made up of 24 physics lectures and 23 economics lectures. The average number of annotated segments for the physics lectures was 6, whereas it was 7.1 for the economics lectures. Table 1 shows the new dataset's relevant statistics.

3.2 Segmentation Models

The performance of six competitive models from the literature was compared, with regards to organising MOOCs text content: C99 [4]; UI [28]; LCSEG [11]; MCS [16]; BSEG [7]; STM [6]. All models are explained in Section 2. The publicly available executable given by the authors was employed in all cases, except for LCSEG[3].

[3]This software needs a copyright license from `http://www.cs.columbia.edu/nlp/tools.cgi#LCseg`

	#Lect	#Segments Per Lect	#Total Segments	#Total Words	#Sentences
Physics	24	6	144	260k	18k
Economics	23	7.1	172	212k	15k
Overall	47	6.5	316	472k	33k

Table 1: Lecture Corpus Statistics.

	Physics		Economics	
Text Segmenter	P_k	WD	P_k	WD
C99	0.429	0.433	0.419	0.426
UI	0.426	0.442	0.425	0.435
LCSEG	0.387	0.394	0.356	0.388
MCS	0.439	0.446	0.378	0.383
BSEG	0.364	0.385	0.313	0.334
BSEG+DC	**0.359**	**0.379**	**0.309**	**0.328**
STM	0.372	0.396	0.311	0.330

Table 2: Results of the comparison between segmentation models: WD denotes WindowDiff. Both metrics are penalties, so lower scores indicate better performance.

The paper's specified parameter values [11] were used in the case of LCSEG. MCS needs parameter settings to be tuned on a development set. The corpus of this study does not incorporate development sets, and as a result the tuning was undertaken with the configuration given by the author on the lecture transcript corpus [16]. On the other hand, C99 and UI do not need parameter tuning and can be used without any modification [4, 28]. BSEG also do not need any parameter tuning, but priors are re-estimated, as noted in the paper [7]. The STM model 10 randomly initialised Gibbs chains were used, where every chain ran for $30,000$ iterations, with $25,000$ for burn-in. Following this, 200 samples used the discount parameter $a = 0.2$, and $\lambda_0 = \lambda_1 = 0.1$ and the Dirichlet prior is $\alpha = 0.2$ and $\gamma = 0.01$. In all experiments, the number of segments is assumed to have been given beforehand.

3.3 Evaluation Metrics

All experiments are evaluated with regards to the widely utilised P_k [1] and WindowDiff (WD) [19] metrics. Both metrics run a window throughout a document, and evaluate if the sentences on the edges of the window were suitably segmented with regards to one another. WD is stricter because it needs the number of intervening segments between the two sentences to be exactly the same in both the hypothesised and reference segmentations, whereas P_k only checks if the two sentences are in the same segment. P_k and WD are penalties, so lower values show superior performance. [10] has provided the evaluation source code that was being used.

4. RESULTS AND DISCUSSION

The different performances of the six segmenters using P_k and WD values are shown in table 2. Overall, superior results across the two disciplines were seen in the BSEG model, especially with discourse cues, and the gain and fails of each model across the two disciplines were described. It should be highlighted that these models show better performance using P_k and produce less improvement on the WD metric. This is explained in Section 3.

Notably, the output of the MCS model, which produces segmentation as a graph cut problem, for the physics lectures yields 0.439 P_k, which is worse off compared to more straightforward similarity-based models, such as the C99 and LCSEG. Other models, such as UI, which do not specifically depend on pairwise similarity analysis, have better performance ($P_k = 0.426$) in physics lectures, when compared to MCS. UI calculates a better segmentation performance by estimating alterations to the language model predictions through various partitions, as described in Section 2. On the other hand, economics lectures differed, as MCS had superior performance ($P_k = 0.378$) compared to both C99 and UI, which yielded $P_k = 0.419$ and $P_k = 0.425$ respectively. This is due to the difference in distributional properties of the physics lectures, which were not coherent in their thematic shifts and thus caused a level of distributional differences.

A further note from Table 2 with regards to the LCSEG model was that it had superior performance on P_k metric for both disciplines ($P_k = 0.387$ in physics and $P_k = 0.356$ in economics), compared to all other models used with the exception of BSEG and STM. STM achieved favourable performance, especially in economics lectures, and attained results close to the BSEG model in physics lectures. This can be attributed once again to the lack of coherence in physics lectures, which results in smooth distributional variations. A substantial and consistent increase is seen through the use of BSEG+DC for all lecture subjects. This can be justified from the existence of discourse cues, as depicted in the results of $P_k = 0.359$ in physics and $P_k = 0.309$ in economics. As spoken language is more impulsive and not as planned as written language, the speaker must inform the listener of any alterations to topic content, through the introduction of subtle cues, and references to prior topics during topical transitions [9].

A further analysis study of discourse cues was undertaken, using the labelled topic boundaries. For every word in the lecture corpus, the number of its occurrences near any topic boundary (with a window size of 5 seconds on either side of the target boundary, inclusive) are counted, and set against those further away. The findings were utilised in the undertaking of the χ^2 significance test. The chi-square test allows the calculation of the significance of the near-against distinct-statistics by comparing with the overall statistics, where the null hypothesis is assumed. The word with an χ^2 value in opposition to the hypothesis under 0.01-level confidence (the rejection criterion is $\chi^2 \geq 6.635$) were chosen. Table 4 shows discourse cues sorted by chi-squared value, where bold denotes the common cues of both disciplines. The corpus was manually examined to find these automatically selected discourse cues, and it was discovered that these cues establish linguistic expressions, as in the study by [27] on summarisation task. An example of this is the cue "*topic*", which is part of one expression, such as "*The topic of this lecture is*" or a very different expression, like "*Let's move to another topic*". These expressions can obviously show the function and the purpose of the discourse, and thus show the pedagogical element of this segment. However, current topic

	Near	Distant	Near	Distant
today	14	53	29	92
Other	15254	244776	17838	193820

Table 3: $\chi^2 = 24.73$ in Physics and 35.82 in Economics.

Physics		Economics	
DC	χ^2	DC	χ^2
topic	110.25	**talk**	140.07
last	105.78	about	126.15
okay	51.89	wanted	97.61
now	37.88	**lecture**	67.97
next	32.05	let	66.97
today	24.73	**we**	65.46
alright	22.52	move	59.17
lecure	18.84	**today**	35.83
we	11.40	conclude	29.07
talk	8.68	start	21.58

Table 4: Automatically selected discourse cues (DC), sorted by chi-squared χ^2 value at the level of $p < 0.01$. Boldface indicates that these cues are common across disciplines.

segmentation models do not account for these expressions, possibly because of the fact that these models lack conversational analysis. Additional research is required to examine this aspect, including the induction of these expressions in the segmentation model and the possibility of using an automatic method to identify and extract these expressions, such as in the study by [15] on the extraction of expressions from student essays.

5. CONCLUSION AND FUTURE WORKS

The application of topic segmentation models for the automatic organisation of MOOCs resources has been presented above. The manual analysis of these resources shows that their structure is centred on both pedagogical and topical aspects, and so a new corpus has been established based on this scenario, through two different domains. The study employs the different features of the topic segmentation models in order to compare the results. The outcomes show that the topic segmentation model which utilised linguistic cues (e.g. *today*, *okay*) had the highest results. An important element for future research is the automatic detection and extraction of linguistic expressions, which are used to show various purposes and functions in discourse, in order to be able to involve them in the topic segmentation model. It can be hypothesised that this type of model would have superior performance in the representation of MOOCs learning units.

6. REFERENCES

[1] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1-3):177–210, Feb. 1999.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] F. Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of NAACL'00*, pages 26–33, 2000.

[4] F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP'01*, pages 109–117, 2001.

[5] R. Correia, N. Mamede, J. Baptista, and M. Eskenazi. Using the crowd to annotate metadiscursive acts. *In Proceedings 10th Joint ISO-ACL SIGSEM*, page 102, 2014.

[6] L. Du, W. L. Buntine, and M. Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, pages 190–200, 2013.

[7] J. Eisenstein and R. Barzilay. Bayesian unsupervised topic segmentation. *Proceedings of EMNLP'08*, page 334, 2008.

[8] J. Eisenstein, R. Barzilay, and R. Davis. Gestural cohesion for topic segmentation. In *Proceedings of ACL-08: HLT*, pages 852–860, 2008.

[9] J. Flowerdew and L. Miller. The teaching of academic listening comprehension and the question of authenticity, 1997.

[10] C. Fournier. Evaluating Text Segmentation using Boundary Edit Distance. In *Proceedings of ACL'13*, 2013.

[11] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. Discourse segmentation of multi-party conversation. In *Proceedings of ACL'03*, pages 562–569, 2003.

[12] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, Mar. 1997.

[13] X. Huang, F. Peng, D. Schuurmans, N. Cercone, and S. E. Robertson. Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, 6(3-4):333–362, 2003.

[14] N. Kokhlikyan, A. Waibel, Y. Zhang, and J. Y. Zhang. Measuring the structural importance through rhetorical structure index. In *HLT-NAACL*, 2013.

[15] N. Madnani, M. Heilman, J. Tetreault, and M. Chodorow. Identifying high-level organizational elements in argumentative discourse. In *Proceedings of NAACL'12: HLT*, pages 20–28, 2012.

[16] I. Malioutov and R. Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings ACL '06*, pages 25–32, 2006.

[17] H. Misra, F. Yvon, J. M. Jose, and O. Cappe. Text segmentation via topic modeling: An analytical study. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1553–1556, 2009.

[18] R. J. Passonneau and D. J. Litman. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139, Mar. 1997.

[19] L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, Mar. 2002.

[20] M. Purver, T. L. Griffiths, K. P. Körding, and J. B. Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of COLING-ACL'06*, pages 17–24, 2006.

[21] S. Ramamurti. Fundamental of physics 1 (yale university: Open yale courses). `http://oyc.yale.edu/physics/phys-200#overview`, 2006. (Accessed December 20, 2014), License: Creative Commons BY-NC-SA.

[22] M. Riedl and C. Biemann. Topictiling: A text segmentation algorithm based on lda. In *Proceedings of ACL'12 Student Research Workshop*, pages 37–42, 2012.

[23] O. Rodriguez. The concept of openness behind c and x-moocs (massive open online courses). *Open Praxis*, 5(1):67–73, 2013.

[24] K. Sathiyamurthy and T. V. Geetha. Automatic organization and generation of presentation slides for e-learning. *Int. J. Distance Educ. Technol.*, 10(3):35–52, July 2012.

[25] R. J. Shiller. Financial markets (yale university: Open yale courses). `http://oyc.yale.edu/economics/econ-252-11`, 2011. (Accessed December 22, 2014), License: Creative Commons BY-NC-SA.

[26] Q. Sun, R. Li, D. Luo, and X. Wu. Text segmentation with lda-based fisher kernel. In *Proceedings of ACL'08: Short Papers*, pages 269–272, 2008.

[27] S. Teufel and M. Moens. Summarizing scientific articles - experiments with relevance and rhetorical status. *Computational Linguistics*, 28:2002, 2002.

[28] M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of ACL'01*, pages 499–506, 2001.

How High School, College, and Online Students Differentially Engage with an Interactive Digital Textbook

Jeremy Warner[1], John Doorenbos[2], Bradley N. Miller[2], Philip J. Guo[1]

[1] University of Rochester, Rochester, New York, USA
[2] Luther College, Decorah, Iowa, USA

ABSTRACT

Digital textbooks have been growing popular as a lower-cost and more interactive alternative to paper books. Despite the recent rise in adoption, little is known about how people use these resources. Prior studies have investigated student perceptions of digital textbooks in the classroom via interviews and surveys but have not quantified actual usage patterns. We present, to our knowledge, the first large-scale quantitative study of digital textbook usage. We mined 6.8 million log events from over 43,000 people interacting with *How To Think Like a Computer Scientist*, one of the most widely-used Web-based textbooks for learning computer programming. We compared engagement patterns among three populations: high school students, college students, and online website viewers. We discovered that people made extensive use of interactive components such as executing code and answering multiple-choice questions, engaged for longer when taking high school or college courses, and frequently viewed textbook sections out of order.

Keywords

Digital textbooks; student engagement; server log data mining

Categories and Subject Descriptors

H.5.1. [**Information Interfaces and Presentation (e.g. HCI)**]: Multimedia Information Systems

1. INTRODUCTION

Digital textbooks have grown popular in the past decade as more students gain access to laptop computers, tablet devices, and broadband Internet. Some of their claimed benefits over paper textbooks include lower cost, lighter physical weight, full-text search, electronic note-taking, and better accessibility for sight-impaired students via text-to-speech [4]. As the costs of paper textbooks continue to rise, university professors are adopting digital alternatives to save money for their students [13]. Governments are pushing for widespread adoption of digital textbooks at the K-12 level as well. For instance, in his 2011 State of the Union address, U.S. President Barack Obama challenged all K-12 schools to adopt digital textbooks by 2016, and the FCC Chairman and Secretary of Education followed up with a plan to implement this vision [12]. The publishing industry has responded to recent events by converting many of their paper textbooks into digital formats. By some estimates, digital textbook sales will be a $1.5 billion business and account for over 25% of all new textbook sales by 2016 [11]. In parallel, universities [1], non-profits, and independent volunteers [8] are developing freely-available digital textbooks.

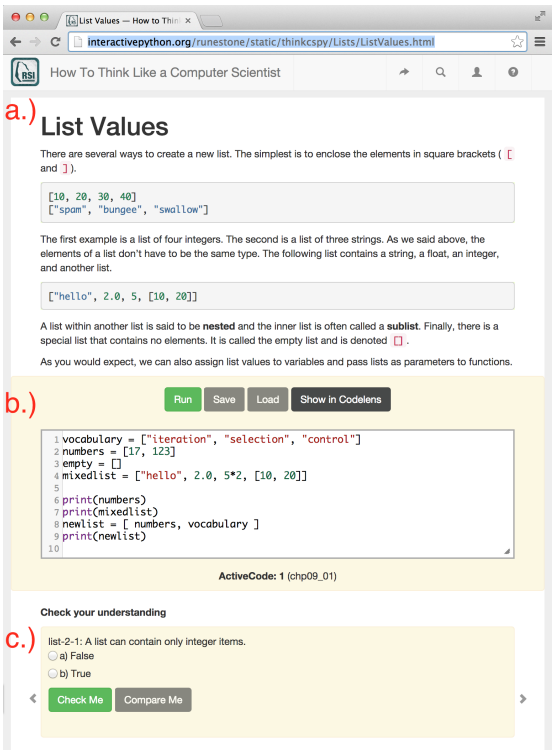


Figure 1: *How To Think Like a Computer Scientist* [8] is a Web-based interactive digital textbook for learning computer programming. A user can: a.) read text, b.) edit and run Python code to see outputs, and c.) answer multiple-choice questions.

Aside from classroom use, online digital textbooks are a form of educational technology similar to MOOCs. Anyone with a computer and Internet connection can learn topics ranging from computer programming [8, 10] to math using digital textbooks. In recent years, many researchers have studied how students use MOOCs [3, 6], but to our knowledge, there has never been an analogous large-scale study of digital textbook usage. Given the growing prominence of digital textbooks, it is important to understand how stu-

dents use them in a variety of educational settings, and how that could inform the design of the next generation of digital textbooks.

This paper contributes, to our knowledge, the first large-scale study of how students use an interactive digital textbook. We studied *How To Think Like a Computer Scientist* [8], a Web-based digital textbook for learning computer programming (Figure 1). We analyzed two years of server logs containing 6.8 million events from 43,416 students. This data is far larger, more diverse, more precise, and finer-grained than prior digital textbook studies that relied on questionnaires sent on university campuses [2, 9, 13].

Specifically, we quantified how students navigated through the textbook and engaged with interactive components such as live code and multiple-choice questions. We segmented students into three populations: those taking a high school course, a college course, and those visiting the public textbook website. These comprise the three main populations of textbook readers. We investigated three sets of research questions: 1.) How much does each population engage with interactive components of the textbook? 2.) When do people in each population access the textbook, and for how long do they persist before quitting? 3.) How do readers navigate non-linearly and skip around when accessing textbook contents?

The first generation of digital textbooks were simply paper books converted into electronic formats such as PDF. The current generation features interactive topic-specific widgets (Figure 1) but does not take advantage of the scale afforded by tens of thousands of online readers. This study is one step toward providing data to inform the design of the next generation of digital textbooks, which can leverage such data to assist students, instructors, and book authors.

2. RELATED WORK

Researchers have studied student attitudes toward digital textbooks in the classroom, with mixed findings. Questionnaire studies of 446 students in the University of Cape Town in South Africa [13] and of 5,000 business school students across 127 U.K. universities [9] found high self-reported enthusiasm for adopting digital textbooks. In contrast, a survey of 662 students across five California State University campuses found that only 1/3 were satisfied with digital textbooks and only 1/2 felt they were easy to use [2]. Prior studies were all done on non-interactive digital textbooks, comparing them to nearly-identical paper counterparts. And they all relied on questionnaires and exam results but did not analyze log data on actual textbook usage. To our knowledge, we are the first to study an interactive digital textbook in-the-wild in a large-scale online setting. Our sample contains 43,416 students from around the world, which is one to two orders of magnitude more students than prior studies.

3. METHODOLOGY

We studied usage patterns of *How To Think Like a Computer Scientist* [8], a widely-used Web-based digital textbook for learning introductory computer programming. This textbook is viewable online for free at `http://interactivepython.org/`. Figure 1 shows how it intersperses textual content, snippets of editable and runnable Python code, and multiple-choice questions. This digital textbook shares similarities with computer programming MOOCs. Both feature multiple-choice questions and runnable Python code as interactive components. However, unlike a MOOC, the main pedagogical modality here is text rather than video. Also, registration is not mandatory. Readers can register with a free account to save their code and track personal analytics, but this is an open resource that anyone can access on the Web. Finally, there is no notion of a fixed course schedule with, say, weekly releases of new materials like there is in some MOOCs. All textbook materials are always present, which supports self-paced learning.

We mined the server logs from June 2012 to 2014, fetching 6,834,244 events from 43,416 students. Each event has the following fields:

- **Timestamp** – server time in the U.S. Central Time Zone
- **Student type** – High School, College, Open (public website)
- **Student ID** – either a registered username or an IP address
- **Event type** – Page load, Run code, Code error, Viz interaction (Python code visualization), or Multiple-choice attempt
- **Textbook location** – the chapter and sub-chapter to which this event belongs (e.g., chapter 5, sub-chapter 3).

Event types: The *Event type* field has one of the following values:

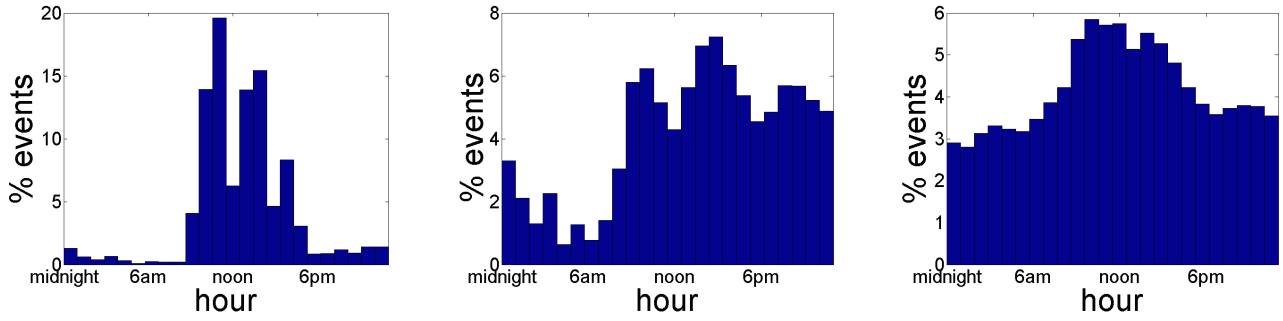
- **Page load** – Load a webpage, which displays the content for a specific sub-chapter of the textbook
- **Run code** – Press the “Run" button to run a piece of Python code, and the code executes successfully (Figure 1b)
- **Code error** – Press the “Run" button to run a piece of Python code, but the code has a syntax or runtime error
- **Viz interaction** – Interact with a Python code visualization widget by taking one step forward or backward in the embedded visual single-step debugger tool [5]
- **Multiple-choice attempt** – Attempt to answer a multiple-choice question within a webpage (Figure 1c)

Non-Linear Navigation: We define a ***backjump*** as any consecutive pair of events for one student where the first occurred in chapter n and the second in chapter m, where $n > m$. A ***sub-backjump*** is either a regular backjump, or a pair of events in the same chapter that went from sub-chapter n to sub-chapter m, where $n > m$. We define skip and sub-skip similarly. A ***skip*** is any consecutive pair of events where a student jumped from chapter n to chapter m, where $m > n+1$. Note that we use $n+1$ because simply going to the next chapter is ordinary sequential navigation, not a skip. A ***sub-skip*** is either a regular skip, or a pair of events in the same chapter that went from sub-chapter n to sub-chapter m, where $m > n+1$. The intuition behind these metrics is that if a student navigated through the textbook in a perfectly sequential fashion, starting with chapter 1, sub-chapter 1, and ending with the final sub-chapter of chapter 15, then they would have zero backjumps or skips. Thus, backjumps and skips indicate non-linear navigation.

4. FINDINGS

4.1 Engagement with Interactive Components

Most students actively engaged with the interactive components rather than just passively reading. Figure 2 shows that page loads accounted for only around 10% of total events. If students had simply been using this textbook as a static reference, then *all* events would have been page loads. By far the most common event type was attempting to run Python code. *Run code* and *Code error* events comprise around three quarters of total events. Recall that pieces of Python code are embedded throughout the textbook (Figure 1b.). Some are complete working examples that can be run verbatim without triggering errors, while others are incomplete snippets that students must complete as an exercise. For all three populations, attempting multiple-choice problems and interacting with code visualizations were about as common as page loads, which again indicates that students did not just passively read the book.



(a) High School (b) College (c) Open

Figure 3: Distributions of events throughout the day, recorded as server time in the U.S. Central Time Zone.

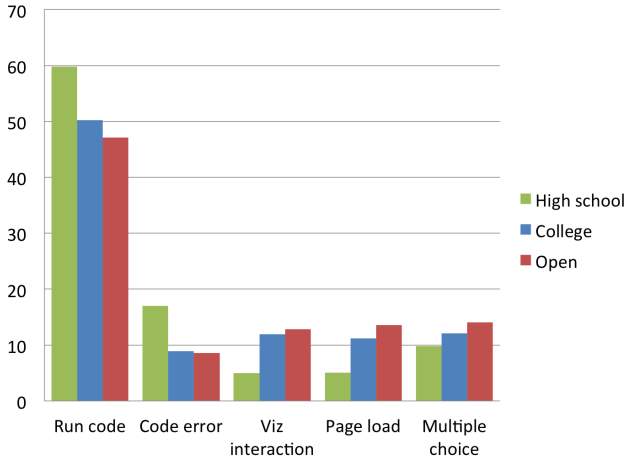


Figure 2: Percentages of total events by type.

4.2 Writing, Running, and Debugging Code

Figure 2 shows that high school students ran the most code, with ∼10% more *Run code* events and twice as many *Code error* events than college and open. Also, for high school students, 22% of total code run attempts resulted in an error, versus only 15% for college and open. High school students made, on average, 112 errors per student, versus 35 errors per student for college and 12 for open.

One interpretation is that high school students made more errors because they were less experienced at coding, but we do not have the data to support this claim. Since this is an introductory textbook, presumably the college and open students also did not have much prior coding experience. A more likely interpretation is that the high school students used this textbook in a more structured and instructor-guided manner than college and open. We have anecdotal evidence from high school teachers who sent emails to the textbook creators requesting technical support that many intended to use this strictly within their classrooms. A typical use case is a teacher directing students to spend the class period reading through a sub-chapter and attempting to do all of the code-related exercises. The teacher would then walk around the classroom and help students debug their faulty code. Thus, high school students ran more code and persisted in debugging, fixing their errors, and re-running possibly because an instructor was present in the classroom.

In contrast, college and open students are usually less supervised. College instructors typically assign readings from a textbook but do not monitor students as closely as high school teachers do. Since running code and attempting multiple-choice problems are ungraded formative exercises, students can work on them at their leisure. Open students might be self-directed learners with little to no supervision. Thus, they make fewer code errors (12 per student) not necessarily because they are better at coding, but simply because they might give up after seeing an error and not persist in fixing it.

4.3 Activity Levels by Time of Day

Visualizing activity levels by time of day confirms that high school students mostly use this textbook in class during school hours, while college and open students use it throughout the day. Figure 3 shows the distribution of event times. The majority of high school activity occurs between school hours of 9am to 4pm, with a sharp dip at noontime for lunch. This pattern indicates in-class usage, supervised by a teacher. In contrast, college activity occurs evenly throughout most waking hours from 8am to midnight.

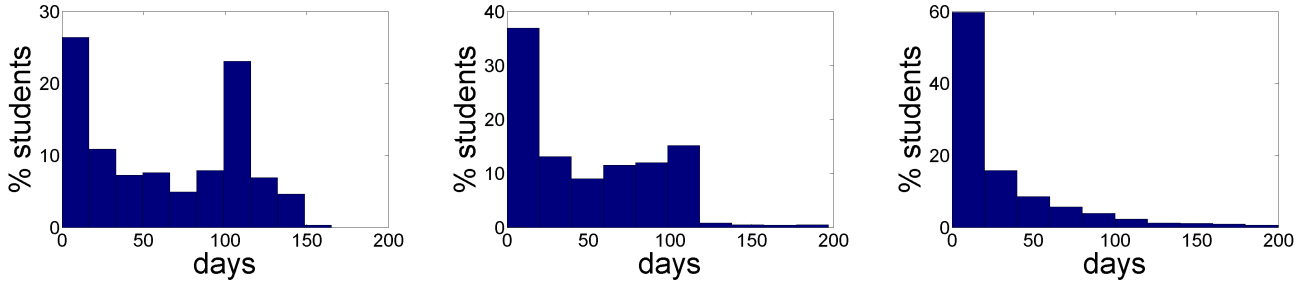
Note that the event timestamp is the server's time (U.S. Central Time Zone), so it does not take the student's local time zone into account. However, by geolocating IP addresses of high school and college students, we found that the majority with a geolocatable IP were from the U.S. and Canada (89% of high school and 94% of college students), so the true time for those students lies within a few hours of the U.S. Central Time Zone.

Whereas high school and college students came mostly from the U.S. and Canada, the open student population was much more international. Only 57% of open students were from the U.S. or Canada, and many came from countries such as Australia, New Zealand, the U.K., and India. Unsurprisingly, those are all English-speaking countries, since this textbook is in English. The presence of many international students explains the relatively even levels of activity throughout the day and night in Figure 3c, although there is still a spike during mid-day in the U.S. and Canada.

4.4 Engagement Duration

For how long does each student engage with the textbook before quitting? We quantified engagement duration by calculating the difference between the first and last event times for each student. Figure 4 plots the distributions for all three student types. High school and college students engaged for up to a semester (∼ 105 days) because they used the textbook as part of a course. The high school spike at around 105 days is much more pronounced than the college one, which could be a result of greater teacher supervision.

In contrast, the open population engagement drops off sharply in a long-tail-like distribution, which mirrors the high initial dropout



(a) High School (mean=62, median=62) (b) College (mean=48, median=39) (c) Open (mean=28, median=13)

Figure 4: Distributions of how many days each student was actively engaging with the textbook, split by student type.

Student Type	Backjumps		Sub-Backjumps	
	mean	median	mean	median
High School	41.0	2	58.4	11
College	13.2	2	21.8	4
Open	3.8	0	6.1	1
	Skips		**Sub-Skips**	
	mean	median	mean	median
High School	38.5	4	67.3	13
College	13.1	1	27.4	7
Open	4.3	1	9.2	3

Table 1: Non-linear navigation statistics for all student types.

rates in MOOCs [3, 7]. Half of the open students used the textbook for less than two weeks. However, unlike many MOOCs, which incrementally release new course materials on a weekly basis, all of the material in this textbook is always available. Thus, it is possible for self-directed learners in the open population to engage for a week or two, learn what they want, and then leave. Thus, semester-long engagement is simply an artifact of formal course schedules.

4.5 Non-Linear Navigation

How frequently did students jump backward to earlier textbook locations or skip forward to latter ones out of sequence? Table 1 summarizes the levels of backjump and skip activity by student type. For all four measures we defined (backjump, sub-backjump, skip, sub-skip), high school students exhibited the most non-linear navigation, followed by college, then open. Even controlling for differing levels of activity per student, high school students perform twice the number of backjumps and skips as college and open students. For instance, 6.2% of all high school events involved backjumps, versus only 3.4% of college and 2.7% of open events.

Non-linear navigation indicates engagement, since it takes more active effort to jump around rather than following the default sequential ordering of the textbook by simply clicking the “Next page" link at the bottom of each page. One explanation for the high numbers of backjumps and skips for high school students is that they are using the textbook in the classroom, so their teacher can proactively direct them to other parts of the textbook as they are trying to solve coding problems. Without other people present in-person to guide or direct one’s learning, it is easier to default back to the more passive style of reading through the textbook in a linear way.

Another interpretation is that high school and college students navigate non-linearly to review materials when studying for exams. A related study of non-linear navigation in MOOCs showed that students often backjumped from exam pages back to earlier lecture pages [6]. In contrast, open students might be self-studying without taking a graded course, so they do not need to review as much.

5. REFERENCES

[1] Open SUNY Textbooks – `http://opensuny.org/`.

[2] Baek, E.-O., and Monaghan, J. Journey to textbook affordability: An investigation of students’ use of eTextbooks at multiple campuses. *The International Review of Research in Open and Distance Learning 14*, 3 (2013).

[3] Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., and Seaton, D. T. Studying learning in the worldwide classroom: Research into edX’s first MOOC. *Research and Practice in Assessment 8* (Summer 2013).

[4] Courduff, J. Digital Textbooks and Students with Special Needs – `http://goo.gl/JGTaA9`, Accessed: Oct, 2014.

[5] Guo, P. J. Online Python Tutor: Embeddable Web-based Program Visualization for CS Education. SIGCSE ’13, ACM (2013), 579–584.

[6] Guo, P. J., and Reinecke, K. Demographic differences in how students navigate through MOOCs. L@S ’14, ACM (New York, NY, USA, 2014), 21–30.

[7] Kizilcec, R. F., Piech, C., and Schneider, E. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. LAK ’13 (2013), 170–179.

[8] Miller, B. N., and Ranum, D. L. Beyond PDF and ePub: Toward an interactive textbook. In *Proceedings of ITiCSE* (2012), 150–155.

[9] Nicholas, D., Rowlands, I., and Jamali, H. R. E-textbook use, information seeking behaviour and its impact: Case study business and management. *Journal of Information Science 36*, 2 (Apr. 2010), 263–280.

[10] Pritchard, D., and Vasiga, T. CS Circles: An In-browser Python Course for Beginners. SIGCSE ’13, ACM (New York, NY, USA, 2013), 591–596.

[11] Reynolds, R. Trends influencing the growth of digital textbooks in us higher education. *Publishing Research Quarterly 27*, 2 (2011), 178–187.

[12] Usdan, J., and Gottheimer, J. FCC Chairman: Digital Textbooks to All Students in Five Years. `http://goo.gl/VJ9NA0`. Accessed: Oct, 2014.

[13] van Heerden, M., Ophoff, J., and Van Belle, J.-P. Are university students ready to dump their textbooks? A survey on student attitudes towards e-readers and tablet computers. *Int’l Jour. Cyber Ethics in Education 2*, 3 (2012), 15–44.

Modeling Exercise Relationships in E-Learning: A Unified Approach

Haw-Shiuan Chang, Hwai-Jung Hsu, Kuan-Ta Chen
Institute of Information Science, Academia Sinica, Taipei, Taiwan
{ken77921, hjhsu, swc}@iis.sinica.edu.tw

ABSTRACT

In an e-learning system, relationships between a large amount of exercises are complex and multi-dimensional; measuring the relationships and arranging curriculums accordingly used to be time consuming and costly tasks which require either enormous log collection or large-scale human annotations. Moreover, accurately quantifying the relationships is difficult because there are too many factors which affect our measurement based on the data, such as the ability of exercise takers and the subject bias of annotators. To overcome these challenges, we propose a unified model that extracts information from both human annotations and usage log using regression analysis. The proposed model is applied to quantify the *similarity*, *difficulty*, and *prerequisite* relationships between every two exercises in a curriculum. As a case study, we collaborate with Junyi Academy, a popular e-learning platform similar to Khan Academy, and infer the pairwise relationships of 370 exercises in its mathematics curriculum. We show that the model can predict exercise relationships as well as an expert does with human annotations of a few sample exercise pairs (2% in our experiments). We expect the introduction of the proposed unified model can improve the relationships among exercises and learning pathways of students in other e-learning platforms.

Keywords

Exercise relationships, Prerequisite, Curriculum, Human annotations, Regression Analysis, Khan Academy

1. INTRODUCTION

Estimating relationships between items has a wide range of applications in educational data mining (EDM). For example, curriculum arrangement [2, 5] and adaptive testing [6, 9] are often based on the estimations of difficulty and prerequisite relationships between courses, knowledge components, or exercises. Furthermore, estimating the similarity and prerequisite relationships between exercises can improve the quality of knowledge components [12, 13] and student modeling [3, 1, 4]. In this paper, we focus on studying the relationships of exercises (i.e., complete question units), which can facilitate personalized education in the future.

Meanwhile, in large and dynamic e-learning websites, manually organizing the growing number of exercises becomes more and more difficult. For instance, Junyi Academy[1], an e-learning platform in Taiwan similar to Khan Academy[2]. Junyi Academy provides over 300 interactive exercises for its mathematics curriculum, which is visualized by the knowledge tree as shown in Figure 1. We can see that there have been many complex prerequisite links in the knowledge tree, so it is very time consuming to manually validate how appropriate the prerequisite links are and whether there are better ways to arrange the links of the exercises. Moreover, the instructors need to consider hundreds of exercise candidates when determining the prerequisites for a new exercise.

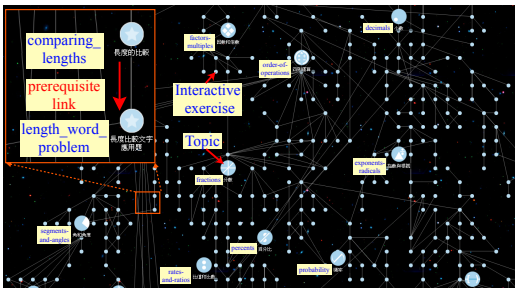


Figure 1: Part of the knowledge map on Junyi Academy. To visualize the prerequisite structure, the knowledge tree is laid out in a 2D plane called knowledge map.

Based on exercise taking log, researchers discover the relationships through item response theory (IRT) [10], inferring Bayesian model of students [3, 12, 1, 4], factor analysis [8], association rule learning [5], assuming a known Q-matrix [13], or assuming students would perform better after they have taken prerequisite or similar exercises [12, 11, 15], etc. Most of the aforementioned data-driven methods develop a specific learning algorithm for estimating a specific relationship between exercises. The learning algorithms usually require a large amount of log data so as to simultaneously infer all latent factors affecting our observation in data, such as relationships of exercises and capability of every student over time. However, data in some e-learning platforms might not be sufficient to accurately profile various behaviors of every student. As a result, the estimation of relationships between exercises might be misleading in a new system with only a small amount of usage log [16, 10].

On the other hand, the collected data are often noisy [1] and have different statistical characteristics in different systems, which might violate the assumptions made by a data-driven model. For example, many e-learning websites, such as Khan Academy and Junyi Academy, allow learners to browse any exercise without actually answering them. In fact, around 70% of the first answers are correct for the first problem of each mathematical exercise on Junyi Academy, which shows that learners tend to skip exercises they cannot answer. The freedom of selecting exercises would degrade the performances of purely data-driven approaches on more difficult exercises with less responses [16], and also cause challenges to identify the difficulty and prerequisite

[1]Junyi Academy (`http://www.junyiacademy.org/`) is established in 2012 on the basis of the open-source code released by Khan Academy.
[2]`https://www.khanacademy.org/`

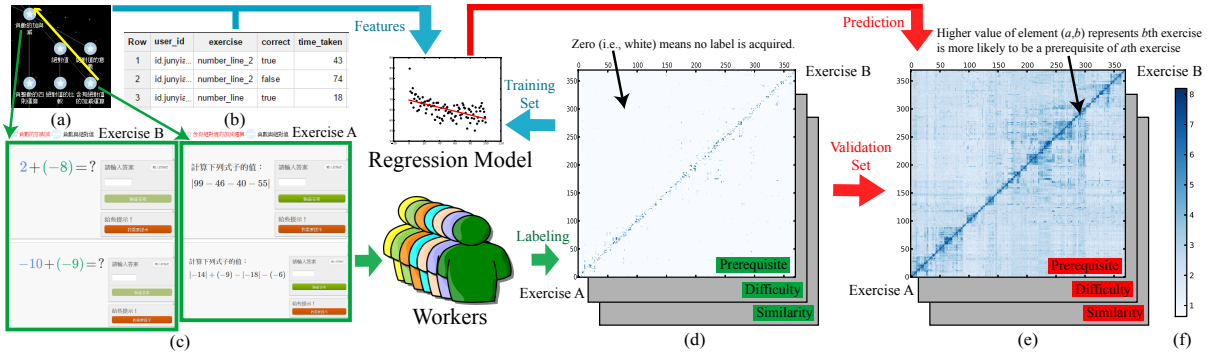


Figure 2: The proposed work flow. (a) A screen shot of the local knowledge map, (b) examples of usage log, (c) a example of exercise pairs, (d) the sparse similarity matrices labeled by workers, (e) the dense similarity matrices predicted by a regression model, and (f) the color code of (d) and (e).

relationships between exercises (See details in Sec. 2.3).

To solve the challenges, we advocate a hybrid method which integrates the power of crowdsourcing and machine learning as [14] did for finding prerequisite relationships among documents. As illustrated in Figure 2, we first quantify the *similarity*, *difficulty*, and *prerequisite* relationships of mathematical exercise pairs using crowd wisdom. Then, we characterize each exercise pair by various types of features extracted from the user practice log and website contents. Given labels and features, a regression model can be trained to predict relationships of every exercise pair. Finally, collected labels can be used to quantitatively evaluate both the prediction of machines and humans. Our experiments show that predictions generated by the proposed models are closer to the crowd consensus (i.e., average opinions of workers) than most of individuals' ratings.

2. RELATIONSHIP DISCOVERY

2.1 Label Collection

As previously discussed, the exercise relationships are hard to define objectively from usage log. Recently, Wauters et al. [16] pointed out that as more annotators judge difficulty of each exercise, their average score converges to a more steady value, which is highly correlated with the difficulty inferred by IRT model. Therefore, if we collect more subjective labels with high quality, their average responses are more representative (i.e., more likely to be agreed by most learners and instructors) and less sensitive to subject bias.

To collect high-quality labels from wide range of people, we divide the task of comparing exercise relationships into several questionnaires and apply several quality control methods. The method includes mathematical ability qualification, malicious workers detection by checking the elapsed time and the variances of their responses in each questionnaire, and outlier filtering using crowd consensus as [7] did.

At each section of questionnaires, we consecutively compare an exercise A with 1–7 other exercises which might be related to A. Note that potentially related exercises are paired according to student modeling and knowledge tree in Figure 1, and the order of comparisons is randomly determined. An example of comparison could be seen in Figure 2(c).

Any target relationship of exercise pairs could be quantified by a specific question. In this work, we ask the workers to choose the 1–9 score for the following questions, which query about *similarity*, *difficulty*, and *prerequisite* relationships of each exercise pair (A and B), respectively.

- How similar is the knowledge required for answering these two exercises?
- How much more difficult is exercise B compared to exercise A, where a higher score means B is more difficult than A and a score of 5 indicates that they have the same difficulty?
- After students learned to correctly answer exercise B, how appropriate is utilizing exercise A to deepen the students' knowledge on the topic step by step?

2.2 Feature Extraction

To automatically predict the relationships, we extract the usage log from Oct. 2012 to July 2014 on Junyi Academy, which contains over 10 million answering records from over 100 thousand users. When describing relationships between exercise A and exercise B, we extract the potentially helpful features from usage log and cluster them into 6 categories:
(i) ***Student Modeling (4 features)*** is extracted based on the practice history of each student. To be more specific, the student is modeled by applying random forest regressor to predict his/her accuracy on every exercise which has not been done by the student. Then, we compute original and normalized feature importance of log data in B for predicting students' accuracy in answering A, and the corresponding importance of A for the prediction of B.
(ii) ***Answering Time Duration (6 features)*** includes the difference between the average answering time duration of A and that of B (i.e., $(time\ for\ A)-(time\ for\ B)$), the logarithm difference of their average answering time duration (i.e., $\log(time\ for\ A)-\log(time\ for\ B)$), the difference and the logarithm difference of their answering time duration on the average of users' correct answers, and on the average of users' first correct answers of the exercises.
(iii) ***#Problems Taken in Exercises (4 features)*** (# means *the number of*) includes the difference and the logarithm difference between #total problems taken in A and B, the difference and the logarithm difference of #problems which are answered correctly in A and B.
(iv) ***Answering Accuracy (6 features)*** includes the difference and the logarithm difference between accuracy of A and that of B on the average of users' first, last, and all answers in the exercises, where the accuracy is defined by $\frac{\#correct\ answers}{\#total\ answers}$. Note that we only count the first answer of each learner in the same problem.
(v) ***#User Taking Exercises (3 features)*** includes the difference and the logarithm difference between #users taking A and that of B, and the Euclidean distance between #users vectors of A and that of B. The ith element in the #users vector of A records the #users who have done exercise i correctly before A.
(vi) ***User Answering Orders (6 features)*** include #users who practice A before B (denoted as $\#U[A \to B]$), #users who do B before A ($\#U[B \to A]$), $\frac{\#U[A \to B]}{\#U[A \to B]+\#U[B \to A]}$, #correct answers for A before answering B ($\#C[A \to B]$), the corresponding #answers for B before A ($\#C[B \to A]$), and $\frac{\#C[A \to B]}{\#C[A \to B]+\#C[B \to A]}$.

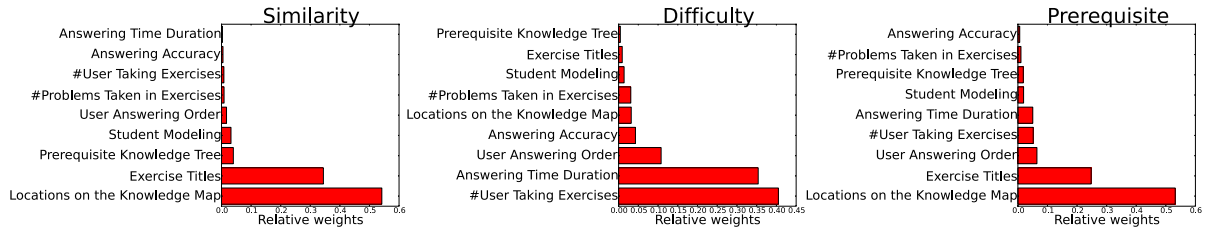


Figure 3: The feature importance for predicting relationships on Junyi Academy. The red bar of each category means the summation of all the feature importance in the category, and symbol # represents *the number of*.

As pointed out in [6, 5, 10, 8, 13], different types of tags on exercises or courses labeled by experts are useful information for determining their relationships. Therefore, we additionally extract exercise-related information from website contents on Junyi Academy, which can be grouped into following 3 categories:
(i) ***Prerequisite Knowledge Tree (5 features)*** includes whether B is a parent of A in the knowledge tree (i.e., the directed acyclic graph), whether B is a sibling of A, distance between A and B in the directed acyclic graph, and the corresponding distances after reversing and removing the direction of every edge in the graph.
(ii) ***Locations on the Knowledge Map (3 features)*** include Euclidean distance between A and B on the knowledge map, and coordinate difference between A and B on x-axis and y-axis in the knowledge map (e.g., the length and the coordinate vector of the yellow arrow in Figure 2(a)).
(iii) ***Exercise Titles (3 features)*** include edit distances of Chinese and English titles between A and B, and summation of the minimal edit distances among English words in their titles.

2.3 Relationship Prediction

Given the features and relationship labels, we formulate the relationship prediction task as a regression analysis. In Sec. 3, we use the collected labels to experiment on the effects of using different regression algorithms. To know the effectiveness of our 40 dimension features, we show the importance of feature categories which are determined by random forest regressor in Figure 3.

Compared with *Answering Accuracy*, *#User Taking Exercises* is a much better type of features for predicting the difficulty difference of exercises, because learners tend to skip exercises they cannot answer as we mentioned in Sec. 1. For the similarity and prerequisite relationships, the *Locations on the Knowledge Map* are the strongest type of features for the tasks, while the *Prerequisite Knowledge Tree* surprisingly has relatively low feature importance. An explanation for the observation is that instructors usually maintain similar exercises in close distance on the knowledge map, which are often good prerequisite candidates for each other. However, when they manually assign the prerequisite links in the knowledge tree, the graph needs to be kept sparse to ensure the clarity and simplicity of its layout.

Figure 3 also illustrates that the information contained in the *Exercise Titles* is much more correlated with the prerequisite relationships on Junyi Academy than features based on *Student Modeling* and *Answering Accuracy*, of which the analysis is extensively studied by many previous works such as [3, 12, 1]. Therefore, it would be interesting to investigate whether the observation is still valid in other platforms which probably have different rules of naming titles or of recommending exercises to learners.

3. EXPERIMENTS

Our proposed method is evaluated in the exercise system of Junyi Academy. To prevent scarce usage log skewing the statistical distribution of our features, we exclude the exercises which are answered by less than 100 users. The remaining 370 exercises of interest are randomly divided into two sets: the training set containing 240 exercises, and the testing set with 130 exercises. On average, each exercise of interest in training set is paired with 4.7 other exercises where around 10% of exercises are randomly selected, and each one in testing set is paired with 6.3 other exercises where the percentage of randomly selected exercises reaches around 30% to verify our generalization capability.

To evaluate how good humans and machines perform, one of metrics we adopt is relative squared error (RSE), which is defined as $\frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$, where $\hat{y}_i$ and y_i are our prediction and the ground truth for a relationship of exercise pair i, respectively, and $\bar{y}$ is the mean of y_i over all i. In addition, we transform every score of exercise relationships into its rank, and compare the similarity between the ranks from the predicted scores and the ranks from the ground truth scores. Then, we evaluate the predicted rank by Spearman's ρ and Kendall τ rank correlation coefficients.

3.1 Performance of Workers

After excluding malicious and unqualified workers, we hire 3 teachers, 8 online workers, and 43 people to work in the lab. All workers in the lab are at least graduated from senior high school, and most of them have a college degree. Each exercise pair in the training set are labeled 6.6 times on average by total 51 normal workers, and teachers are asked to score all the exercise pairs in the testing set. For the interest of the consistency between judgements from crowd consensus (i.e., the average scores from all workers) and that from experts, we also ask 2 among 3 teachers to label every pair in the training set. The total costs of collecting above labels are around 1,000 USD.

Manually quantifying the relationships between mathematical exercises is a demanding cognitive task, which requires a certain level of skills in abstract reasoning. Using the average of ratings from all workers (including teachers) as our ground truth, we first evaluate the performances of recruited workers and whether teachers (i.e., experts) perform better in the tasks. The results in the training set are presented in Table 1. Note that smaller RSE and larger rank coefficients indicate better performances. From Table 1, it is clear that the performance of workers (including experts) measured by RSE is significantly lower than the ones measured by rank coefficients compared with the performances of machines. The results illustrate that workers' annotations often contain systematic subject bias (i.e., workers tend to rate every query higher or lower than most of other people), so averaging scores rated by multiple workers is an effective way to improve the labeling quality for the task.

Table 1: Performance comparisons of different methods in the training set of Junyi Academy using cross validation, and best performances among regressors are highlighted in bold font.

Methods			Similarity			Difficulty			Prerequisite		
			RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ
Humans	An Normal Worker	Range	0.193–1.124	0.188–0.854	0.208–0.750	0.492–3.235	0.063–0.820	0.050–0.747	0.316–2.381	0.000–0.813	-0.007–0.725
		Mean	0.574	0.598	0.524	1.096	0.516	0.439	0.986	0.458	0.387
	A Teacher	Range	0.493–0.543	0.648–0.718	0.560–0.625	0.619–0.741	0.625–0.634	0.539–0.540	0.858–1.054	0.571–0.684	0.504–0.594
		Mean	0.518	0.683	0.593	0.680	0.630	0.539	0.956	0.638	0.549
Regressors	Linear Regression		0.370	0.658	0.567	0.470	0.593	0.504	0.424	**0.624**	**0.541**
	nu-SVR		0.349	**0.683**	**0.594**	0.483	**0.611**	**0.526**	0.402	0.611	0.520
	Random Forest Regression		0.320	0.662	0.575	0.493	0.576	0.493	0.376	0.608	0.516
	GBR		**0.288**	0.680	0.590	**0.453**	0.610	0.521	**0.346**	0.600	0.514
Features (GBR)	w/o KT and KM		0.311	0.681	0.589	0.474	0.626	0.532	0.378	0.607	0.515
	w/o KT, KM, and ET		0.433	0.607	0.521	0.472	0.642	0.546	0.472	0.567	0.478
	w/ SM, AA, UN, and PT		0.548	0.516	0.438	0.502	0.610	0.516	0.595	0.463	0.377
	w/ SM and AA		0.598	0.524	0.446	0.632	0.486	0.400	0.666	0.417	0.346
	w/ KT		0.674	0.463	0.418	0.869	0.448	0.382	0.717	0.360	0.318

Table 2: Performance comparisons of different methods in the testing set of Junyi Academy. Note that the meaning of all abbreviations is the same as Table 1.

Methods			Similarity			Difficulty			Prerequisite		
			RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ
Humans	A Teacher	Range	0.200–0.300	0.764–0.848	0.656–0.757	0.398–0.474	0.732–0.791	0.629–0.696	0.322–0.467	0.696–0.764	0.583–0.665
		Mean	0.235	0.814	0.719	0.427	0.762	0.661	0.406	0.721	0.617
Regressors	GBR		0.269	0.786	0.678	0.553	0.580	0.476	0.311	0.771	0.660

3.2 Prediction Accuracy

For the training set, we evaluate our prediction by 5-fold cross validation, and Table 1 compares the resulting outputs generated by different regression models and different subsets of features. The table summarizes the results of five regression algorithms including linear regression, nu support vector regression (nu-SVR), random forest regression, and gradient boosting regression (GBR). Compared with teachers' ratings in the training set, our approach can generate competitive performances measured by rank coefficients while having lower RSE, especially for more complex regressors such as the random forest or gradient boosting algorithms. This means that after being trained by collected labels, machines could predict exercise relationships closer to crowd consensus than most of the individuals. Note that to make the comparison fair, we round all of the scores predicted by machines into integers between 1–9.

In Table 1, we also provide control experiments on different types of features using gradient boosting regression. There might not be the knowledge tree (KT) and the knowledge map (KM) in other interactive learning environments, so we first present the performance without related categories of features. The results show that removing KT and KM can still produces competitive performances, but the performance would decrease by a margin if we further remove more features such as *Exercise Titles* (ET), *User Answering Orders*, *Answering Time Duration*, *User Numbers* (UN), and *#Problems Taken in Exercises* (PT), *Student Modeling* (SM), and *Answering Accuracy* (AA).

In order to verify our generalization ability across different types of annotators, we train the regression models on the training set (mostly labeled by normal workers) and evaluate their performance on testing set (labeled by teachers). As shown in Table 2, the performances of regression models are still very promising. Note that the exercise pairs in the testing set are only rated by 3 teachers whose labels have larger impact on ground truth, so the real performances of experts might be worse than this estimation.

4. CONCLUSIONS

The relationships of exercises are important for curriculum arrangement of e-learning platforms. In this work, we demonstrate that the relationships can be quantified by subjective labeling and predicted by regression models. The experiments on Junyi Academy show that the quality of predicted relationships are competitive against teachers' labels.

References

[1] E. Brunskill. Estimating prerequisite structure from noisy data. In *EDM*, 2011.

[2] E. Brunskill and S. J. Russell. RAPID: A reachable anytime planner for imprecisely-sensed domains. In *UAI*, 2010.

[3] C. Carmona, E. MillÃąn, J.-L. P. de-la Cruz, M. Trella, and R. Conejo. Introducing prerequisite relations in a multi-layered bayesian student model. In *User Modeling*, 2005.

[4] M. C. Desmarais. Performance comparison of item-to-item skills models with the IRT single latent trait model. In *UMAP*, 2011.

[5] T.-C. Hsieh and T.-I. Wang. A mining-based approach on discovering courses pattern for constructing suitable learning path. *Expert Syst. Appl.*, 2010.

[6] C. Koutsojannis, G. Beligiannis, I. Hatzilygeroudis, and C. Papavlasopoulos. Using a hybrid AI approach for exercise difficulty level adaptation. *IJCEELL*, 2007.

[7] B. Lakshminarayanan and Y. W. Teh. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. *CoRR*, 2013.

[8] A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk. Tag-aware ordinal sparse factor analysis for learning and content analytics. In *EDM*, 2013.

[9] D. Lynch and C. P. Howlin. Real world usage of an adaptive testing algorithm to uncover latent knowledge. In *ICERI*, 2014.

[10] M. L. Nguyen, S. C. Hui, and A. C. M. Fong. Content-based collaborative filtering for question difficulty calibration. In *PRICAI*, 2012.

[11] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. In *EDM*, 2009.

[12] P. I. Pavlik, H. Cen, L. Wu, and K. R. Koedinger. Using item-type performance covariance to improve the skill model of an existing tutor. In *EDM*, 2008.

[13] R. Scheines, E. Silver, and I. Goldin. Discovering prerequisite relationships among knowledge components. In *EDM*, 2014.

[14] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012.

[15] A. Vuong, T. Nixon, and B. Towle. A method for finding prerequisites within a curriculum. In *EDM*, 2011.

[16] K. Wauters, P. Desmet, and W. van den Noortgate. Acquiring item difficulty estimates: a collaborative effort of data and judgment. In *EDM*, 2011.

Using knowledge components for collaborative filtering in adaptive tutoring systems

Peter Halkier Nicolajsen
University of Copenhagen
srb816@alumni.ku.dk

Barbara Plank
University of Copenhagen
bplank@cst.dk

ABSTRACT

In adaptive tutoring systems, accurately assessing the ability of a student is central to prescribing the tasks that best facilitate learning. For the 2010 KDD Cup challenge a data set of logs from the Cognitive Tutor system was made available, and contestants were asked to predict the correctness of a student's attempt to answer questions. A successful approach included a collaborative filtering system which predicted student performance on the basis of the performance of similar students. In this paper, we present an extension of this approach. Rather than finding similar students on the basis of their performance on specific questions, we based our similarity measure on the performance on questions that require the same "knowledge components" (or skills). This approach increases the amount of users with whom it is possible to compare performance, which in turn increases the likelihood of finding similar students. The experiments using our question type-based distance measure yield promising results.

Keywords

Adaptive tutoring systems, collaborative filtering, distance measure

1. INTRODUCTION

Education is getting more expensive, a reason for this is that the spread of technology-based improvements in productivity have been very limited compared to other industries. If technological advances allow the same amount of labor to be more productive, that production will become less expensive. Education is a sector where the amount of *output* (i.e., students taught) per hour of teacher labor has been relatively constant. This means that relative to sectors with more technology-based productivity gains–most sectors–education becomes more expensive. In economics this is referred to as *Baumol's cost disease* [3].

Assessment is an element of teaching that is amongst the most labor intensive and thus calls most for technological advancement. In order to give appropriate feedback, it is necessary for a teacher to have an accurate, and up to date picture of the ability of the students. Assessing students requires attention, which naturally limits the number of students that can be effectively supervised. If assessments could be made more efficient, more of the teacher's time could be spent giving appropriate feedback. An educational technology that is based on this idea is the adaptive tutoring system (ATS). An ATS is a platform that delivers educational materials (e.g. lectures, problems etc.) while assessing the student and—as the student uses the system—adapting the material to best suit each student. One instance of an adaptive tutoring system is Carnegie Learning's Cognitive Tutor. This system is based on the ACT-R model of cognition [1, 2]. Logs of interactions with this system for Algebra courses were made available in the 2010 KDD Cup [7], where the task was to predict student performance based on logs of previous interactions. If we use performance as a proxy for ability, a more accurate performance prediction corresponds to a better ability assessment.

In this paper, we propose to extend the work of Töscher & Jahrer [9] (referred to as TJ). Part of their solution was a k-nearest neighbor system that predicted scores based on a weighted average of the 41 most similar students. Here, we propose using a different distance measure, by looking at the students with highly correlated performance scores on similar problems, rather than on identical problems.

2. ADAPTIVE TUTORING SYSTEMS

The Cognitive Tutor is an adaptive tutoring system that provides practice for different subjects. The system assigns specific problems for the user to rehearse on. The student's performance on these problems then allows the system to suggest the appropriate level of additional problems. Figure 1 shows an example screenshot from the system.

The Cognitive Tutor has been developed on the basis of the ACT-R model of cognition [1, 2]. There are two elements of ACT-R that are particularly relevant to learning. The first element is the idea that all complex knowledge is the combination of smaller, discrete, pieces of knowledge, so-called *knowledge components* (KCs). The second element is that a student improves a KC by rehearsing it often and in different contexts. When using the Cognitive Tutor, a student will acquire some complex knowledge by incrementally rehearsing each of the required KCs. Any subject for which

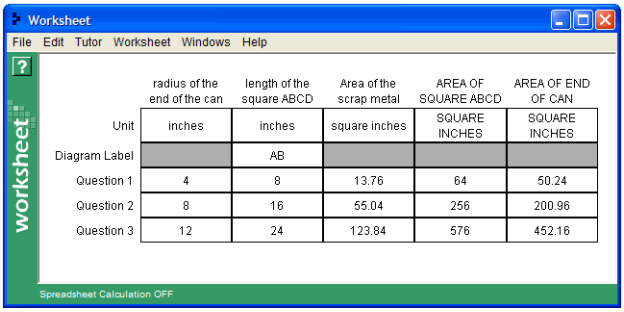


Figure 1: Screen shot from Cognitive Tutor. Each field is one step, while each column consists of three steps that share one knowledge component.

a Cognitive Tutor is implemented, must first be subject to a decomposition analysis, where subject experts identify all of the required KCs, and arrange them in a hierarchy based on the order in which they must be learned. The idea is that, once the subject has been mapped, the student will then be assigned problems that rehearse KCs appropriate to the current level of the student. To assess the student's level, the system keeps track of whether or not the student is able to consistently and correctly, solve problems associated with each KC, in different contexts. For example, being consistently able to solve $4 + 3$ is not the same thing as being consistently able to solve single digit addition. In the course of a session, the system will thus assign several different problems, that require the same KCs.

3. COLLABORATIVE FILTERING FOR ATS

Next, we outline the collaborative filtering approach that was part of [9] (ranking 3rd in the competition) and our proposed extension.

3.1 Töscher and Jahrer

Töscher and Jahrer [9] adopted a collaborative filtering solution, used in the field of recommender systems (e.g., Netflix challenge), and adapted it to the KDD cup challenge. Conceptually the challenges have similarities. The task in the Netflix Prize competition was to recommend movies based on ratings that different users would give different movies, based on the other movies they had rated. The KDD Cup task also required assigning values to different items (steps) for different users based on their previous data. Given these similarities, they proposed a user-based collaborative filtering approach based on the k-nearest neighbor algorithm with correlation shrinkage, described next.

The k-nearest neighbor algorithm found the 41 most similar students for each student, based on how correlated their results were on the basis of the steps they had in common. The prediction is then made on the basis of this group of similar students by using a weighted average (see details below). The stronger a neighbor correlated with the student, the more weight was given to their contribution to the prediction. If the correlation with the whole group was not very strong, the prediction would be corrected toward the student's own mean score. Despite creating the groups based only on correlations in the correct first attempt rate for the different steps, this classifier reached good performance.

The distance measure TJ used was Pearson correlation. Because there was a lot of variation in how many steps each pair of students had in common, the correlation value was transformed to reflect the support for each correlation, giving higher value to correlations based on more common steps. For the sake of consistency, we will use the same terminology as TJ in the algorithm description. They use the terms *students* and *items* to describe the main elements of the model. The items here are the step names. The students s are in the set $\mathbb{S}$, while the steps i are in the set $\mathbb{I}$. The variable to be predicted is whether a student s answered correctly on the first attempt at a step i, is called c_{is}, while the predicted value for this is $\hat{c}_{is}$.

To find the most similar students, the Pearson correlations are calculated between all pairs of students for the steps that both students s_1 and s_2 have answered. The set of steps for s_1 is $\mathbb{I}s_1$, so the set of common steps is $\mathbb{I}s_1s_2 = \mathbb{I}s_1 \cap \mathbb{I}s_2$. Then, the Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i\in\mathbb{I}_{s_1 s_2}}(c_{s_1 i}-\mu_{s_1})(c_{s_2 i}-\mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i\in\mathbb{I}_{s_1 s_2}}(c_{s_1 i}-\mu_{s_1})^2}\sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i\in\mathbb{I}_{s_1 s_2}}(c_{s_2 i}-\mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i\in\mathbb{I}_{s_1 s_2}} c_{s_1 i}$ and $\mu_{s_2} = \frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i\in\mathbb{I}_{s_1 s_2}} c_{s_2 i}$.
To account for the large variability in the number of common steps, they perform a shrinkage transformation that adjusts the correlation by scaling it to the number of common steps $|\ \mathbb{I}_{s1s2}\ |$, this transformation of the correlations is calculated as:

$$\bar{\rho} = \frac{|\ \mathbb{I}_{s_1 s_2}\ | \cdot \rho_{s_1 s_2}}{|\ \mathbb{I}_{s_1 s_2}\ | + \alpha}$$

They set the meta parameter α to a value of 12.9. In the KDD Cup paper [9] they do not describe how they obtain α, but in the Netflix Prize competition paper [8]–where they use an identical shrinkage transformation–they explain that they used a random search method in which they iterate through parameter values selected from a normal distribution, until they find the value that minimizes error. This method is also used to find the other meta-parameters K (set to 41), β (set to 1.5), δ (set to 6.2) and γ (set to -1.9). We here use the same parameters throughout the paper, and leave parameter optimization for future work.
Finally, another transformation is performed on the correlations, in order to minimize the error. The transformation uses the sigmoid function[1]: $\sigma(x) = \frac{1}{1+e^{(-x)}}$ The sigmoid function is then applied to the correlations according to:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

To calculate a predicted score, the scores of the 41 most similar students (k=41) are averaged for the relevant step. Each student's average for the step is then weighted by how strong the correlation is.

$$\tilde{c}_{is} = \frac{\sum \tilde{s}\in\mathbb{S}_i(s;K)\tilde{\rho}_{s\tilde{s}}c_{i\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_i(s;K)\ |\ \tilde{\rho}_{s\tilde{s}}\ |}$$

where $\mathbb{S}_i(s;K)$ is the set of nearest neighbor.
The last element of the algorithm is a final correction of the

[1]Note that the original paper [9] contains a typo, describing the sigmoid function as $\sigma(x) = \frac{1}{1-e^{(-x)}}$

prediction towards the mean score μ_s of student s. This is also necessary in case there is not enough support among the neighbors to make a prediction. The β term ensures that the summed correlation to the neighbors is strong enough that the prediction can be based on it, if the correlation is 0, the prediction will simply be the average score for the student.

3.2 Extension of the approach

The system described in the following is a replication and extension of the k-nearest neighbor model described above. In contrast to the TJ model, we here propose to find similarities based on knowledge components rather than just steps. This idea can be seen as abstracting from concrete question instances to basic concepts of knowledge.

The distance measure used was the correlation between students on correct answer rates for steps sharing the same knowledge component, rather than the same step name. The fact that KCs each represent several step names, means that on average, each pair of students will have more steps on which to be compared. Referring to Figure 1, this would correspond to comparing performance on steps in the same column, rather than on identical steps. In the internal training set the average number of common steps between any pair of students is 40.66, while the average number of common KCs is 52.20. Using this distance measure can be advantageous both by expanding the number of other students with which it is possible to test correlation, and by providing a broader base of problems from which to predict a score.

The procedure for finding the neighbors is the same as above, only with the compared items being different. They are now KC names rather than step names. So the knowledge components KC are in the set $\mathbb{KC}$. The predicted value for the to be predicted CFA (cf. section 4.1) then becomes: $\hat{c}_{KCs}$. The Pearson correlations are again calculated between all pairs of students, this time for the steps that have KCs that both students s_1 and s_2 encounter. The set of KCs for s_1 is $\mathbb{KC}s_1$, so the set of common steps is $\mathbb{KC}s_1s_2 = \mathbb{KC}s_1 \cap \mathbb{KC}s_2$ The Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{i\in\mathbb{KC}_{s_1 s_2}}(c_{s_1 KC}-\mu_{s_1})(c_{s_2 KC}-\mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC\in\mathbb{KC}_{s_1 s_2}}(c_{s_1 KC}-\mu_{s_1})^2}\sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{i\in\mathbb{KC}_{s_1 s_2}}(c_{s_2 KC}-\mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC\in\mathbb{KC}_{s_1 s_2}} c_{s_1 KC}$ and
$\mu_{s_2} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC\in\mathbb{KC}_{s_1 s_2}} c_{s_2 KC}$. The shrinkage transformation is also changed to reflect the number of steps with common KCs:

$$\bar{\rho} = \frac{\mid \mathbb{KC}_{s_1 s_2} \mid \cdot \rho_{s_1 s_2}}{\mid \mathbb{KC}_{s_1 s_2} \mid + \alpha}$$

The correlations again undergo the same sigmoid transformation as in the case of the stepwise algorithm:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

The calculation of the predicted score is altered to use the all of the steps of the most similar students that share a KC with the to be predicted score, again weighted by each neighbor $\tilde{s}$'s correlation to s:

$$\tilde{c}_{KCs} = \frac{\sum \tilde{s}\in\mathbb{S}_{KC}(s;K)\tilde{\rho}_{s\tilde{s}}c_{KC\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_{KC}(s;K) \mid \tilde{\rho}_{s\tilde{s}} \mid}$$

where $\mathbb{S}_{KC}(s;K)$ is the set of nearest neighbors.

4. EXPERIMENTS

4.1 KDD Cup 2010

In 2010 a large amount of log files from the Cognitive Tutor system for algebra was made available for the KDD Cup competition held in conjunction with a data mining conference. These logs contained data on interactions for more than 3,000 students over the course of a school year. Every entry was an interaction of a student with the system. For each student there was an an average of 2700 interactions.

Student ID	Problem Hierarchy	Problem Name	Problem View	Step Name	Knowledge Components	Correct First Attempt
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R1C1		1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R1C2		0
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R2C1		1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R2C2	Identifying units	1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R3C1	Define Variable	1
stu_de2777346	Unit CTA1_01, Section CT/	L1FB12	1	R3C2	Write expressior	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	MoreOrFew	Compare units	0
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	Conversion	Enter unit conve	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	SelectFracti	Select form of or	1

Figure 2: Structure of data, from [5].

The information provided in the data set (see excerpt in Figure 2) included unique identifiers for the student and the interaction, identifiers for the task, information on the success of the student on this interaction, as well as time-stamp information. Every interaction was also marked with an indicator for whether the user solved the step correctly at the first attempt (CFA). The task of the competition was then to predict the "correct first attempt" value of each student for each step, on the basis of the data describing the previous interactions with the system. The step on which the CFA was to be predicted was always drawn from an interaction occurring later than the interactions in the data set.

Since the official test set is not available, we follow standard data splitting practices [10, 4]. In the same way that the organizers had created their test set by taking the last instance of each problem, we created an internal test set by separating out the last two instances of each step within the training set to create an internal test set roughly one tenth the size of the training set. As a result of this split, any step name that occurred fewer than three times was discarded. Ultimately, that left an internal data-set of 6,596,059 training instances, with 662,074 instances in the test set. This internal set then contains 13,128 distinct steps. This also meant that some students with very few lines were discarded, which left 3,079 students.

Due to time constraints it was only possible to test the predictions on a sample of 50 students. Results for the baseline systems are reported on these same 50 students, which means that they are tested on 11,888 rows in the test set (note, results are similar to the entire data set, cf. Section 4.3). The k-nearest neighbor systems still find the 41 most similar students among all 3,079, just like the average based baselines are still calculated from all 3,079 students.

4.2 Evaluation Method

We here use the same evaluation measure as in the KDD cup, i.e., root mean squared error: $RMSE = \sqrt{\frac{\sum(\tilde{c}-c)^2}{n}}$ where $\tilde{c}$ is the predicted score, c is the actual score, and n is the number of predictions.

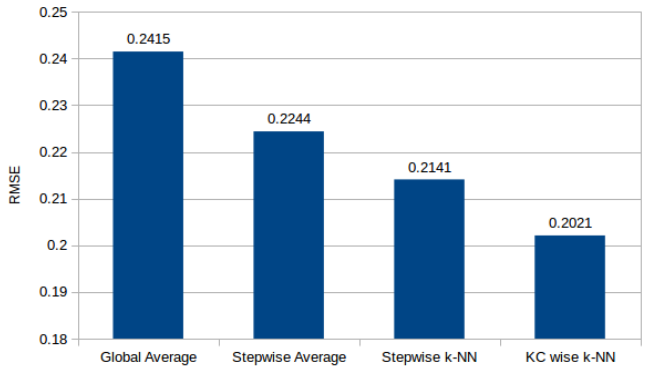


Figure 3: RMSE scores on the 50 student sample from the internal test set.

4.3 Results

Global average baseline. The first, and most basic baseline simply predicts the same score for every problem, 0.8494. The prediction is the average rate of correct first attempts, from the whole training set: $\tilde{c} = \frac{\sum c_{train}}{n}$
For first 50 students in the test set (11,888 predictions) this gave a score of: $RMSE = 0.2415$. For comparison, if we consider the entire test sets (662,074 rows), this system gave a score of $RMSE = 0.2394$.

Stepwise average baseline. The second baseline was already a clear improvement. This system distinguishes between stepnames, and uses the average score for the step in the training set to predict: $\tilde{c_i} = \frac{\sum c_{train_i}}{n_i}$
For first 50 students in the test set this gave a score of: $RMSE = 0.2244$ ($RMSE = 0.2255$ on the full set).

k-nearest neighbor (stepwise) baseline. The third baseline system is the replication of TJ's nearest neighbor system, which makes predictions by taking a weighted average of the scores on the predicted steps for the 41 students with the most similar results in the training set (cf. Section 3.1) This baseline gave further improvement on the second baseline. For the first fifty students in the development set this gave a score of: $RMSE = 0.2141$.

Knowledge component-based k-nearest neighbor system. Our expanded version of the k-nearest neighbor system also predicts on the basis of a weighted average of the scores for the 41 most similar students, but measures proximity on common steps with the same KCs rather than on common steps with the same names. For the first fifty students in the development set this gave a score of: $RMSE = 0.2021$. The results are visualized in Figure 3.

5. RELATED WORK

The 2010 KDD cup received submissions based on a large variety of approaches, many of the highest scoring system being ensemble methods such as [10] (ranking first). Another approach which also accounts for differences between students and problems combines HMMs with bagged decision trees, ranking fourth [6].

6. CONCLUSIONS

We propose to use the performance on similar steps instead of performance on identical steps as a novel distance measure in a collaborative filtering approach to ATS. So far, we only evaluated it on a reduced but reasonably large sample (11,888), but we hypothesize that the prediction error would remain low on the full set, particularly with optimization of hyper-parameters. One potential argument against using KCs is that an expert is needed to decompose the subject material and annotate the KCs. In order to provide learning material, it is necessary have a overview of what the material consists of and the order in which the different elements should be prescribed to best facilitate learning. It would be interesting to automatically learn such a structure, as in fact exploiting latent content is important for improved prediction [4]. However, the aim of this paper is to gauge whether exploiting KC information is sensible, and our preliminary results show that KCs are a potentially valuable source of information. They provide an opportunity to leverage the higher-level structure of the material to gain information about the learning process.

7. REFERENCES

[1] J. Anderson and C. Schunn. *Implications of the ACT-R learning theory: No magic bullets. Advances in instructional psychology.* Educational design and cognitive science, 2000.

[2] J. R. Anderson. ACT: A simple theory of complex cognition. *American Psychologist*, 51(4):355, 1996.

[3] W. J. Baumol. *The next twenty-five years of public choice*, chapter Health care, education and the cost disease: a looming crisis for public choice, pages 17–28. Springer Netherlands, 1993.

[4] S. Cetintas, L. Si, Y. P. Xin, and R. Tzur. Probabilistic latent class models for predicting student performance. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013.

[5] T.-N. Nguyen. *Predicting Student Performance in an Intelligent Tutoring System.* PhD thesis, University of Hildesheim, 2011.

[6] Z. A. Pardos and N. T. Heffernan. Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research W & CP.*, 2010.

[7] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra I 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge., (2010). Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

[8] A. Töscher and M. Jahrer. The bigchaos solution to the netflix prize 2008. *Netflix Prize Report*, 2008.

[9] A. Töscher and M. Jahrer. Collaborative filtering applied to educational data mining. *Proceedings of the KDD Cup 2010 Workshop*, 2010.

[10] H.-F. Yu, H.-Y. Lo, and et al. Feature engineering and classifier ensemble for kdd cup 2010. *Proceedings of the KDD Cup 2010 Workshop*, Feature engineering and classifier ensemble for KDD cup 2010.

Exploring the influence of ICT in online students through data mining tools

Javier Bravo
javier.bravo@udima.es

Sonia Janeth Romero
soniajaneth.romero@udima.es

María Luna
maria.luna@udima.es

Sonia Pamplona
sonia.pamplona@udima.es

Madrid Open University, UDIMA
Carretera de la Coruña 38.500. Vía de servicio 15.
28400, Collado Villalba, Madrid (Spain)

ABSTRACT

The aim of the present work is to evaluate differences according to age in digital competence, usages, and attitude towards ICT in a sample of 1231 online students of a distance university. To fulfill this goal, hypothesis testing, correlation analysis, and data mining techniques were performed on the basis of a 72-item survey. Results showed no strong differences between extreme groups of age. Besides, some interesting correlations between variables and additional information through association rules were found. This study led to better knowledge of online students in order to improve teaching and learning processes.

Keywords

Association rules, ICT attitude, ICT usages, distance education, online education, correlation, Mann-Whitney test, digital competence.

1. INTRODUCTION

During the last decades the proportion of higher education students taking at least one online course has outstandingly increased [1]. A research line developed in the field of e-learning in higher education focuses on the students' access, competences, actions and attitudes towards digital tools and devices and on how those variables are related to learning and well-being. As it is known, distance education, fostered by ICT, increases the variety of learners attending higher studies, creating new challenges for educators and institutions [2]. Specifically there are recent studies on whether the students' age is an important variable. In contrast to the concept of "digital natives" [10] , several studies find no evidences for strong discontinuity of young people on the use and attitudes about digital technology [6] [8]. Nevertheless differences related to age have been found, such us a deeper approach to studying of older students and less time spent using ICT (although this last difference is more noticeable at face-to-face universities than at the distance ones) [5,6].

In this paper we try to address this problem by developing the following objectives: (1) Compare digital competence, uses, and attitude towards ICT between young and students over 50; (2) Analyze relationships between variables in young and students over 50; (3) Obtain additional information about relationships between variables and group of ages by using data mining tools.

This paper is organized as follows. The next section presents a brief selection of related works. The method is described in section three. In section four the results are exhibited. Finally, the section five concludes the paper with discussion and plans for future work.

2. RELATED WORKS

Recently, some research studies were proposed to address the usage of data mining techniques in education especially in association rule mining.

Fattah et al. presented an association rule discovery model to investigate and analyze a university admission system database [3]. The model discovered the relation between students' data and their application status in the university system. The information discovered was very important to the admissions office in the analyzed university because it showed how to filter the applicants with respect to their record in high school.

García et al. described a data mining tool that uses association rule mining and collaborative filtering in order to make recommendations to instructors about how to improve e-learning courses [4]. This tool enables the sharing and scoring of rules discovered by other teachers in similar courses. The work showed and explained some examples of rules discovered in an adaptive web-based course.

Romero et al. explored the extraction of rare association rules when gathering student usage data from a Moodle system [11]. They showed how some specific algorithms, such as Apriori-Inverse and Apriori-Rare, are better at discovering rare-association rules than other non-specific algorithms, such as Apriori-Frequent and Apriori-Infrequent. Finally, they showed how the rules discovered by rare association rule mining algorithms can help the instructor to detect infrequent student behavior/activities in an e-learning environment such as Moodle.

Merceron and Yacef gave an interpretation of two measures of interest through association rules: cosine and added value [9]. In addition, they presented a case study that depicts a standard situation: a LMS that provides additional resources for students as a complement to the face-to-face teaching context. An important conclusion of this work is that common LMS are far from being data mining friendly. Thus, LMS should be enhanced with a special module with good facilities for exploring data.

Kumar and Chadha [7] used association rules mining in discovering the factors that affect assessment in Haryana University (India). They analyzed data for some courses taught in order to measure the students' performance based on factors such as instructor behavior, curriculum design, time schedule and students' interests.

3. METHOD

3.1 Participants, variables and instruments

A total of 1231 students participated voluntarily (with informed consent) in this study, 600 females and 631 males. They were all

students recruited from Madrid Open University in Spain. 63.44% of the sample were studying a Bachelor's degree and 36.56% were Master's students . 40.76% of the students worked in ICT related areas and 57.66% had completed undergraduate studies previously. All the participants were between 18 and 69 years old (mean= 36.01, SD=9.59) and 110 are older than 50 (age 50+ group).

A survey of 72 items was designed to test students' self-reported ICT abilities, uses and attitudes. This survey is divided into four parts: demographical data and academic performance, actions with digital devices (computers, Smartphones, and other digital devices), frequencies of use of ICT tools (digital devices, communications, Moodle, file managing, and other tools) and attitude towards ICT in the process of learning.

(1) Demographical data and academic performance. Students were asked about: age (integer), gender (1=female, 2=male), grade (from 0 to 10), study area (58 values), first enrollment in UDIMA (from 2009-10 to 2014-15), previous degrees, if they work in fields related to ICT and average grade on academic record (retrieved using student identity).

(2) Actions with digital devices. It is composed of 20 items distributed on three scales: Actions with Other Digital Devices (AODD-4 items), Actions with Computers (AC-8 items) and Actions with Smartphones and Tablets (AST-8 items). The format used is 4-point Likert type, from 1 (I cannot do it) to 4 (I can do it and explain it to others). Descriptive results on this block of the instrument are: AODD (min=4; max=16; mean=15.02; SD=1.75); AC (min=10; max=32; mean=28.65; SD=3.96); AST (min=11; max=32; mean=29.04; SD=3.76).

(3) Frequencies of use of ICT tools. It is composed of 25 items distributed on five parts: a) Other Digital Devices (FODD-5 items), b) Communications (FC-5 items), c) Moodle (FM-7 items) d) File Management (FFM-3 items), and e) Other Tools (FOT-4 items). The format used is 4-point Likert type, measuring frequency of use, from 1 (I do not use/do not know) to 4 (Very often). Descriptive results on this block of the instrument are: FDD (min=5; max=20; mean=14.97; SD=3.01); FC (min=7; max=24; mean=17; SD=3.69); FM (min=7; max=28; mean=19.48; SD=4.23), FFM (min=3; max=12; mean=6.32; SD=2.33); FOT (min=4; max=16; mean=6.69; SD=2.46).

(4) Attitude towards ICT in the learning process. It is composed of 24 items distributed on three scales: affective, cognitive and behavioral. The format used is 5-point Likert type from 1 (totally disagree) to 5 (totally agree). Two items on each dimension are inversely rated. The higher test score indicates greater favorable attitude towards the incorporation of ICT in the learning process. Descriptive results shows: min=29; max=120; mean=97.79; SD=13.49. Cronbach Alpha (considering all 24 items of attitude) was 0.89 indicating the high reliability of the test.

3.2 Data analysis

Data analysis included hypothesis testing, correlation and data mining analysis. These are detailed below.

3.2.1 Hypothesis testing and correlation

a) Hypothesis testing. Wilcoxon and Mann-Whitney tests were made to test the hypothesis about differences between extreme ages (24- and 50+).

b) Correlation. Pearson correlation matrix between continuous variables was made in order to evaluate possible associations.

3.2.2 Data Mining Techniques

To complement and provide additional information we used four data mining techniques: OneR, Decision trees (J48), Naïve Bayes and association rules. In this stage it is important to perform the preprocessing phase [12].

a) Preprocessing phase. It is important to note that in this analysis we utilized the whole data. In addition, the items of "Frequencies of use ICT tools" were grouped in four nominal variables: FODD (it contains the sum of five items of "Other Digital Devices"), FC (sum of five items of "Communications"), FM (sum of seven items of "Moodle"), FFM (sum of three items of "File Management"), FOT (sum of 4 items of Other tools). In the same way, the items of "Attitude towards ICT in the learning process" were grouped in the ictAttitude variable. Classification techniques works better with nominal variables. Therefore, age and ictAttitude were discretized to ictAttitude3groups and age4groups respectively. The ictAttitude variable is a continuous variable ranged from 29 to 120. We discretized this variable in three nominal values according to its 33 percentile, 66 percentile, and 99 percentile. Regarding to age variable, it ranged from 18 to 69. This variable was discretized in four values according to its four quartiles. This new variable is called ictAttitude3groups.

b) Selection of variables. In this phase we only use 14 variables: codDegree, gender, firstEnrollment, gradeRound, AODD, AC, AST, FODD, FC, FM, FFM, FOT, age4groups, and ictAttitude3groups. In addition, we utilized the WraperSubsetEval method provided by Weka [13]. This metaselection method selects the most appropriate variables for a data mining technique. This method receives two variables: the selection method and the search method. Since OneR, J48, and Naïve Bayes are classification techniques we indicated to this method to use J48 for selection method (selection mode: 10-fold cross-validation). Also, BestFirst forward method was used for searching method. As a result 8 variables were selected: gender, AODD, AC, FM, FFM, FOT, age4groups, and ictAttitude3groups.

c) Application of techniques. In this phase we utilized three classification techniques and association rules. The classification techniques utilized were: OneR, J48, and Naïve Bayes. We utilized for these techniques the 8 variables listed above (class variable: ictAttitude3groups). In order to select the most appropriate technique we calculated the accuracy (number of correctly classify instances) of each technique. As the size of data was enough to apply the split method, which divides the sample in two parts: training and testing data, we utilized it instead of the cross-validation method. Moreover, we utilized 80% of data for training and 20% for testing. It is well known that if the data size is large enough both methods of dividing the data should give similar accuracies. Concerning association rules we utilized the Apriori algorithm with confidence=0.9 and minimum support=0.1.

4. RESULTS

4.1 Hypothesis testing and correlations

a) Hypothesis testing. First, assumption evaluation was made in order to decide statistical techniques to compare students of extreme ages: above 50 years (50+), and below 24 years (24-). Levene's test shows a lack of homoscedasticity between groups in the variables related with actions: L=20.068, (1, 125), p<.001 for

AODD; L=19.177, (1, 125), p<.001 for AC and L=58.028, (1, 125), p<.001for FDD. The Shapiro-Wilk test shows a lack of normality in all the variables except FC. Due to failure to meet the assumptions we decided to use nonparametric statistic (Mann-Whitney test) to compare both groups. As it can be seen in Table 1, significant differences between both groups (50+ and 24-) were found on actions with other digital devices, actions with Smartphones and tablets, and frequency of use of communication tools. The sum of ranks (see Table 1) indicated high scores on AODD, AST and FC in the group of age 24-.

b) Correlation. The Pearson correlation matrix between continuous variables was calculated for both groups. In the group of age 50+ all the action variables correlated significantly and positively with frequency variables and attitude towards ICT: the lowest positive and significantly correlation was r=.111(AC-FC) and the highest r=.724 (AC-AODD), this result was expected because the more digital competence self-perceived, the more frequent use of ICT tools and the more positive attitude towards ICT use in the learning process. On the other hand, AODD, AST and FC correlated negatively with age (r=-.148, p<.01; r=-.208, p<.01 and r=-.069, p<.05 respectively) indicating as the age increased, the self-perceived competence in actions with digital devices and Smartphones decreased and the use of communication tools was less frequent. The average grade correlated positively with self-perceived competence in actions with computers (r=.136, p<.01). And finally, the number of years studying through distance learning correlated with AC (r=.087, p<.01), FDD (r=.068, p<.05), FM (r=.083, p<.01), attitude (r=.128, p<.01) and age (r=.278, p<.01) indicating that the more years of experience studying online, the higher self-perceived competence of actions with computers and higher frequency of use of digital devices, Moodle, and better attitude toward ICT.

In the group of young students we found less significant correlations, as the number of years studying online did not correlate with any variable, the average grade only correlate positively with AC (r=.197, p<.01), the age correlated inversely with AST (r=-.226, p<.05) and FDD (r=-.307, p<.01). The attitude toward ICT correlated positively with all the actions (r=.205, p<.01 for AODD; r=.221, p<.05 for AC and r=.305, p<.01 for AST) and also with FM (r=.300, p<.01). Finally, the scales of frequencies correlated positively with each other but not with the actions scales.

Table 1. Mann-Whitney U, Wilcoxon W, Z and significance

	ADD	AST	FC
U	5098.500	3429.500	5528.500
W	10454.500	8785.500	10781.500
Z	-4.082	-6.895	-2.438
Sig.	.000	.000	.015
Age	Rank sum	Rank sum	Rank sum
24-	18225.5	17170.5	19894.5
50+	10454.5	11509.5	8785.5

4.2 Data mining techniques

4.2.1 Classification techniques

The accuracies of applying the OneR, J48 and Naïve Bayes technique are as follows: 41.9%, 43.8%, and 42.9%, respectively. For example, J48 classifies correctly in 43.8% of the instances. It is clear that neither of the three techniques has an accuracy greater than 44%. As none of these techniques obtained reliable results, we applied the association rules technique.

4.2.2 Association rules

This trial consisted of using 14 variables with the Apriori algorithm. It is important to highlight that the Apriori algorithm only works with nominal variables. Therefore, the grade variable was removed from the data. The parameters for this algorithm were: minimum support=0.1; confidence=0.9; number of rules=20; instances=1231; attributes=(codDegree, gender, firstEnrollment, AODD, AC, AST, FODD, FC, FM, FFM, FOT, age4groups, ictAttitude3groups).

The result of applying this algorithm is presented in Table 2. It is shown that the Apriori algorithm selected 16 rules. Thus, it is shown that AR6 indicates that students with the best score in actions with computers and best ICT attitude will have the best score in actions with other digital devices. The AR7 and AR14 rules indicate that both genders will have the best score in "actions with other digital devices" if they have the best score in "actions with computers". Both association rules are redundant, since this fact is indicated in AR11. The AR9 contains the variable age. It indicates that the students between second and third quartile of age with the best score in "actions with computers" are experts managing other digital devices.

Regarding the other association rules interesting relations between several variables were found. The first rule relates actions with computers, actions with Smartphones, and actions with other digital devices. Thus, it means that students with the best ICT attitude who demonstrate a high level of actions with computers and Smartphones, will have a high level of actions with other digital devices. The AR2 rule shows a similar relation, but only for male students. The AR3 is informed about a general relation between actions with computers, actions with Smartphones, and actions with other digital devices. A value of 32 indicates a high level of actions with computers and Smartphones, and a value of 16 is also a high level of actions with other digital devices. Thus, a student with a high level of actions with computers and Smartphones, he/she will have a high level of actions with other digital devices. Interesting information is revealed in the AR4 rule, since it relates the first enrollment, action with computers, Smartphones, and with other digital devices. Thus, the students of the 2014-15 year show a high level of action with computers and Smartphones, and also with other digital devices. This association rule is similar to the AR10 and AR15, but with less information. The rules AR5 and AR6 show that students with a high ICT attitude, and a high value in actions with computers or Smartphones, will have a high level in actions with other digital devices. Finally, the AR8 and AR16 rules relate the gender, action with computers and with other digital devices. Consequently, female and male students report the same abilities in actions with Smartphones and other digital devices.

5. CONCLUSIONS

The present study gathered a large sample composed of 1231 online students in a distance university with a range of age from 18 to 69 years. Our results agree to a great extent with other related studies [5][6]. In fact, we did not find enough evidence of strong differences among extreme groups of age, although results showed slight differences in variables related with the frequency

of use and perceived competence with Smartphones and communication tools.

Another interesting conclusion is that attitude towards ICT did not correlate inversely with age, on the contrary, students aged 50+ exhibited positive attitudes towards the implementation of ICT for the learning process. These conclusions lead to better knowledge about students attending online higher education. Therefore, these results should provide improvements in the methodology of the e-Learning courses and foster the utilization of communication tools (less utilized by 50+ students).

This work also showed that data mining techniques can provide complementary information to traditional analysis methods. Although classification techniques did not provide reliable results, since its accuracy was less than 44%, the association rules technique provided deeper information. In fact, the Apriori algorithm obtained 16 association rules. These association rules showed relationships between the following variables: actions with computers, Smartphones and other digital devices, gender, ITC attitude, and first enrollment in UDIMA. This information was not provided by the hypothesis testing, therefore, we have demonstrated that association rules are appropriate to analyze these data.

For future work it will be appropriate to analyze other parameters of the Apriori algorithm that could provide rules with more information. For instance, to test and evaluate other selection methods based on Lift or Leverage is an interesting future line of research [9].

Table 2. Best rules of the Apriori algorithm

Rule		Cov.	Conf.
AR1	AC=32 AST=32 ictAttitude3groups=3 ==> AODD=16	137	1
AR2	gender=2 AC=32 AST=32 ==> AODD=16	221	0.99
AR3	AC=32 AST=32 ==> AODD=16	307	0.99
AR4	firstEnrollment=2014-15 AC=32 AST=32 ==> AODD=16	134	0.99
AR5	AST=32 ictAttitude3groups=3 ==> AODD=16	167	0.98
AR6	AC=32 ictAttitude3groups=3 ==> AODD=16	187	0.97
AR7	gender=1 AC=32 ==> AODD=16	143	0.97
AR8	gender=2 AST=32 ==> AODD=16	264	0.97
AR9	AC=32 age4groups=3 ==> AODD=16	123	0.96
AR10	firstEnrollment=2014-15 AC=32 ==> AODD=16	191	0.96
AR11	AC=32 ==> AODD=16	426	0.96
AR12	AST=32 ==> AODD=16	390	0.95
AR13	gender=2 firstEnrollment=2014-15 AC=32 ==> AODD=16	126	0.95
AR14	gender=2 AC=32 ==> AODD=16	283	0.95
AR15	firstEnrollment=2014-15 AST=32 ==> AODD=16	182	0.93
AR16	gender=1 AST=32 ==> AODD=16	126	0.91

6. ACKNOWLEDGMENTS

This work has been funded by European Commission Lifelong Learning Programme through the AGE50+ project (hash code: 1B8E6E156B13780A).

7. REFERENCES

[1] Allen, I. E., & Seaman, J. 2013. *Changing Course: Ten Years of Tracking Online Education in the United States*. Sloan Consortium. PO Box 1238, Newburyport, MA 01950, USA.

[2] Altbach, P. G., Reisberg, L., & Rumbley, L. E. 2009. Trends in global higher education: Tracking an academic revolution. UNESCO.

[3] Fattah Mashat, A., M.Fouad, M., S. Yu, P. and F. Gharib, T. 2013. Discovery of Association Rules from University Admission System Data. *International Journal of Modern Education and Computer Science* 5,4 (May 2013), 1–7.

[4] García, E., Romero, C., Ventura, S. and de Castro, C. 2011. A collaborative educational association rule mining tool. The Internet and Higher Education 14, 2 (March 2011), 77–88.

[5] Hosein, A., Ramanau, R., & Jones, C. 2010. Learning and living technologies: a longitudinal study of first year students' frequency and competence in the use of ICT. *Learning, media and technology* 35, 4 (2010), 403-418.

[6] Jelfs, A., & Richardson, J. T. 2013. The use of digital technologies across the adult life span in distance education. *British Journal of Educational Technology* 44, 2 (2013), 338-351.

[7] Kumar, V. and Chadha, A. 2012. Mining association rules in student's assessment data. *International Journal of Computer Science Issues* 9, 5 (September 2012), 211–216.

[8] Lai, K. W., & Hong, K. S. 2014. Technology use and learning characteristics of students in higher education: Do generational differences exist?. *British Journal of Educational Technology*. doi:10.1111/bjet.12161

[9] Merceron, A. and Yacef, K. 2008. Interestingness measures for association rules in educational data. *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, Quebec, Canada, June 20-21, 2008). 57–66.

[10] Prensky, M. 2001. Digital natives, digital immigrants part 1. *On the horizon*, *9*, 5 (2001), 1-6.

[11] Romero, C. and Romero, J. 2010. Mining rare association rules from e-learning data. *Proceedings of the 3rd International Conference on Educational Data Mining* (Pittsburgh, PA, USA, June 11-13, 2010). 171–180.

[12] Vialardi, C., Bravo, J., & Ortigosa, A. 2008. Improving AEH Courses through Log Analysis. *Journal of Universal Computer Science* 14,17 (Sept. 2008), 2777-2798.

[13] Witten, I.H., Frank, E., & Hall, M.A. 2011. Data Mining: practical machine learning tools and techniques. Morgan Kaufmann Publishers. Burlington, MA 01803, USA.

Understanding Revision Planning in Peer-Reviewed Writing

Alok Baikadi
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
baikadi@pitt.edu

Christian Schunn
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
schunn@pitt.edu

Kevin Ashley
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
ashley@pitt.edu

ABSTRACT

Revision is a core writing skill that presents challenges to both novice and expert writers. Within the context of peer review, peer feedback has the potential to provide rich guidance for revision, especially when making content-level changes. However, authors must review and evaluate each piece of feedback for meaningful critiques that can be applied to further drafts. In this work, we analyzed several factors that influenced students' decisions to fix or ignore comments they received. We found that feedback on content dimensions, as well as critical remarks by both the reviewers, and by the authors regarding papers they reviewed, were correlated with the amount of revisions made between drafts.

Keywords

peer review, revision, writing instruction

1. INTRODUCTION

Revision has long been seen as one of the cornerstones of effective writing [6]. Practicing revision has been shown to not only improve the produced writing, but also help on first drafts of future writings [9]. One of the discriminators between expert and novice writers is how they approach revision. While both groups often make many surface-level edits, such as spelling, grammar, and stylistic revisions [4, 14, 17, 2], expert writers often make a higher proportion of content-level edits than do novices [3].

By using a peer-review approach, students were able to employ more strategic revision strategies given peer feedback [10], make fewer surface-level changes [15], and add more details in their writing [12], especially when peers provide justification for their feedback [7]. Once feedback is received, it is not always implemented in future drafts [5, 2]. Sometimes students indicate an intention to implement meaningful changes but do not follow through with the intent [4]. Checklists [16] and revision memos [1] have been used to focus students' revisions on important aspects of their writing.

Within peer-review, it has not been clear how often students forget about the feedback received during revision, rather than make a choice to disregard the feedback. An accurate model of revision behavior could allow a teacher or intelligent system to intervene for students who require additional support. Diagnostic information could also be presented to the teacher as to what kinds of comments are being made, how they are being received, and what sorts of revisions to expect in future drafts. An effective model could also be used to provide hints to students, about how their feedback may be received as reviewers and which comments provide meaningful feedback for revision as an author.

In this work, we investigated this decision within a web-based peer-review application. We present a revision planning application designed to scaffold the process of evaluating feedback received in the peer-review process. We analyzed their responses within the system in order to better understand why some comments may be addressed while others are ignored. Critical comments about the content of the paper, rather than the surface aspects, were more likely to be included in their revision plan, and were more highly correlated with changes in the second draft.

2. REVISION PLANNING CORPUS

Web-based, computer-supported peer review has been shown to be an effective tool for improving students' writing skills. Students still need support, however, in organizing the reviews they receive and planning how to revise their own papers. This paper describes a revision environment that helps students to cluster and prioritize reviewers' suggestions, develop a plan for revision their papers, and make note of lessons learned about writing for future use. We report here about students' experiences in using the tool in an undergraduate Cognitive Psychology course.

2.1 SWoRD Peer Review

Scaffolded Writing and Rewriting in the Disciplines (SWoRD) is a web-based reciprocal peer review system. Over the past 12 years, it has been used by over thirty-five thousand students across grade levels and across a variety of academic disciplines. The peer review process within SWoRD takes place in three phases: An Authoring phase, a Review phase, and a Revision phase. In the first phase, students submit a response to an instructor-provided writing prompt. Students may either enter text into the web interface, or upload a pre-existing document in order to submit their assignments. During the Review phase, students are presented

with the grading rubric and comment prompts the instructor has provided along with the submitted document. The student reads the document and provides written feedback for each evaluative dimension, as well as numerical scores on a seven-point rating scale. In the final phase, students receive the feedback and scores generated by their peers. The process is then repeated for the second draft.

2.2 Revision Planning

During the course of peer review, students have the opportunity to learn from both giving and receiving feedback. During the review process, students are asked to critically evaluate a peer's submission on the same rubric with which their own writing will be judged. While reviewing, students may notice aspects of their peers' submissions that they can incorporate into their own work. Many revisions occurred when the student both recognized it in a peer's work, as well as received feedback on the same topic from their peers [13].

To support this process, the Revision Planning system has two components. The Lessons Learned page, shown in Figure 1, is available to the student during the reviewing process. It encourages them to make observations on the papers they are reading, and how that may be applied to their own document. They are able to identify the observation as a good idea that they'd like to consider for their revisions, or a problem that they would like to avoid.

The Revision Planner, shown in Figure 2, allows students to consider how they would address each comment they receive from their peers. For each comment, they can elect to ignore it or fix it. If they choose to fix it, they can then assign a priority and make notes on what the revision will be. If they choose to ignore it, they can select a reason from a drop-down menu, or add text to explain why it is being ignored. Both the Revision Planner and the Lessons Learned are visible during revision. The system can also generate checklist that the students can use during their revisions.

2.3 Data Collection

The data were collected from 75 college students in an introductory Cognitive Psychology course, all of whom had completed a required writing seminar prior to enrollment. The students were asked to write a 1,000 word article imitating a newspaper style that connects topics discussed in class with their everyday lives. The rubric included several dimensions regarding the communicativeness of the article, such as its interestingness, word choice, and quality of writing, and several about the course content, such as the relevance and accuracy of the concepts introduced in the course. Of the 75 students, 60 completed the Revision Plan, and 44 completed the Lessons Learned. A second draft was submitted, and subjected to the same peer review process, without additional revision planning support.

Each student was asked to review four peer submissions during the revision phase. In addition, students were allowed to perform bonus reviewing for extra credit. For each review (n=297), we collected 10 numerical scores, which were separated among the five evaluation dimensions. Students were required to write at least one textual comment for each dimension, though they could provide up to five different textual comments for a single dimension. For each textual comment the student received (n=1822), we recorded the decision to "Fix" or "Ignore" the comment, a discretized reason for marking the comment as "Ignore" when provided, as well as the text of the intended revision and priority.

3. REVISION PLANNING BEHAVIOR

Using the data described above, we investigated four main research questions: (1) what factors influenced the students' decision to fix or ignore a comment that they received, (2) what were the reasons that students gave for ignoring a comment, (3) how is the process of revision planning within Anonymous related to the amount of revisions between the first and second drafts, and (4) how are the observations made on the Lessons Learned page related to the amount of revisions between the first and second drafts.

3.1 Fix and Ignore Decisions

For each comment, we calculated a score given by the reviewer by averaging all scores for the comment's dimension. If there were multiple comments within the same dimension, they received the same score. The score serves as a proxy for how critical a comment is. A dimension type (content or communication) was derived by grouping the three communication-related dimensions together, and grouping the other two dimensions as content. Prior work [17] has indicated that content feedback is more likely to result in content revisions. The length of the comment was computed in number of characters, following the intuition that longer comments are more likely to contain useful feedback.

On average, students elected to mark only 44% of their comments as "Fix" (sd=0.21). We performed a logistic regression analysis, shown in Table 1, to determine which factors influenced the decision to fix or ignore a comment. All three factors were shown to have a significant main effect, and there was a marginally significant interaction between the score and the dimension type.

Table 1: Logistic Regression for Fix Decisions

Variable	Coefficient	z-Value	p-Value
Score	0.74	-5.61	< 0.001
Content Dimension	3.71	238	0.018
Comment Length	1.00	9.73	< 0.001
Score x Dimension	0.85	-1.74	0.082

On average, students elected to fix approximately 40% of their comments in the Communication dimensions, compared to 48% of their comments in the Content dimensions. Comments marked as "Fix" were on average longer (mean=283) than those marked as "Ignore" (mean = 188). Figure 3 shows the proportion of comments fixed by score and the type of dimension.

3.2 Ignore Reasons

There were seven categories of reasons students could select when they ignored a comment: no critique was given, the student disagreed with the comment, the comment was already mentioned elsewhere, the comment is only praise, the comment is only a summary, the comment was confusing, and other. Figure 4 shows the distribution of categories that were provided if any was given. Since the "Summary",

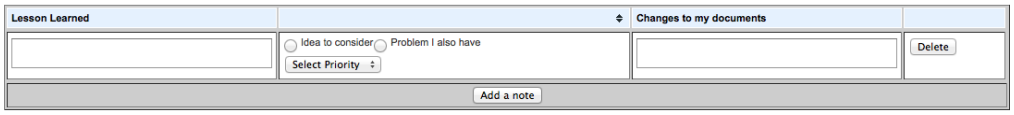


Figure 1: Lessons Learned Page

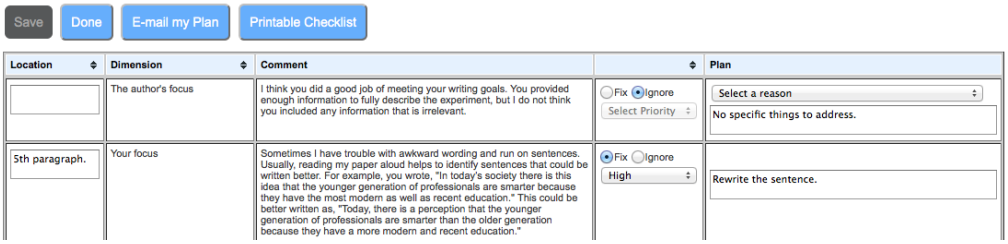


Figure 2: Revision Planning interface

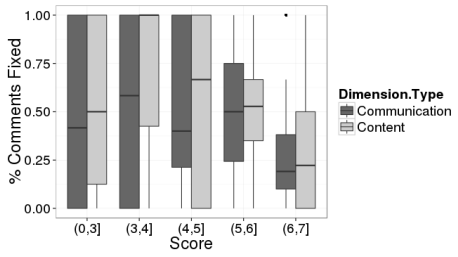


Figure 3: Percent of Comments marked as Fix by Score and Dimension Type

"Confusing", and "Other" categories occurred relatively infrequently, we omitted them from further analyses.

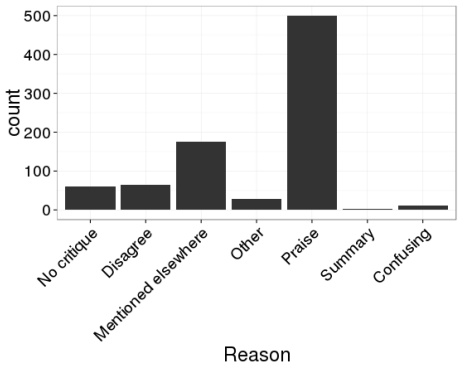


Figure 4: Distribution of Ignore Reasons

Table 2 shows the results of a multinomial logistic regression analysis, relative to the "Praise" category, to determine which factors influenced the category. There was a significant effect of the score for distinguishing all categories. In addition, there was a significant effect of dimension type for the "Mentioned Elsewhere" category, and a significant effect for the comment length on both the "Disagree" category, and the "Mentioned Elsewhere" category.

Table 2: Logistic Regression for Ignore Reasons

Reason	Content	Score	Comment Length
No Critique	0.62	0.66 **	0.99
Disagree	1.38	0.35 ***	1.01 ***
Elsewhere	0.46 *	0.32 ***	1.01 ***

3.3 Revision Planning and Revision

In order to measure changes in the drafts, all submissions were first converted to a plain text format. Both drafts were then segmented using the Stanford Parser [11] and compared using CompareSuite, a software package for analyzing text documents. Edits were compared at the sentence level by calculating how many sentences were added, deleted, or modified [8]. These numbers were then compared against the number of sentences in the first draft to calculate the amount of change between drafts. There was a weak correlation (r=0.20) between the proportion of comments labeled as "Fix", and the amount changed. However, there was a moderate relationship with the proportion of Content comments labeled as "Fix" (r=0.37), while there was no relationship (r=0.10) with the proportion of Communication comments labeled as "Fix".

3.4 Lessons Learned and Revision

For students who completed the lessons learned (n=44), we also investigated how the different types of observations effected the revisions. Students made an average of 2.8 (sd=1.96) observations (See Figure 5). Pearson correlations showed that neither the number of good observations (r=-0.14) nor the total number of observations (r=-0.039) was correlated with the amount of revisions. However, the number of critical observations was moderately correlated with the amount of revision (r=0.31).

4. CONCLUSIONS AND FUTURE WORK

In this work, we analyzed several factors that influenced students' decisions to fix or ignore a comment they received. The content dimensions offered the most insight into the revision behavior of the students. Content comments were more likely to be marked as a comment to fix, and when they were fixed were more highly correlated with the amount of

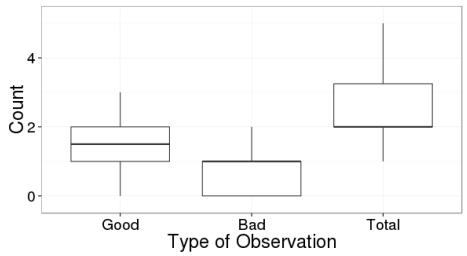


Figure 5: Distribution of Lessons Learned

revision done between drafts. While we did not analyze the comments made by the students, the students were specifically instructed to give feedback on the breadth and accuracy of the domain content in the content dimensions. In addition, lower scoring comments were more likely to be marked as fix or marked as "Mentioned Elsewhere", especially in the content dimensions. This latter selection indicates that the students intended to fix these issues, but had recognized them either through their own experience or through other comments, and were therefore more willing to ignore the specific feedback in those comments. Comments that were highly scored were more likely to be praise or otherwise lack critique. Relatively few comments were ignored because the students disagreed with the feedback received, and those tended to be at the extremes of the scores. The fact that few comments were ignored due to a disagreement with the critique, and the fact that critical observations made from other peers' submissions were more highly correlated with the amount of revision between drafts suggests that students benefit more from critical analysis of the papers they have both read and written.

One of the discriminating features between novice and expert writers is how they approach revision, particularly in terms of how often they revise for deeper meaning. While our results show correlations to the amount of revision done, further analysis will need to be done regarding the quality of the revisions. While comment length was surprisingly informative, it is an extremely shallow measure of the comment text. There are also many other factors that could inform the students' decisions on how to approach the comments they get, such as the helpfulness rating, and the relative strength of the writing skills between the author and reviewer. In terms of student revision process, a more fine-grained analysis of whether students fixed the comments they said they would, could be instrumental in supporting the effectiveness of the scaffolding mechanisms. It was also somewhat surprising that critical observations of peers' papers in the Lessons Learned were also correlated with more revision. One question raised by this observation is whether students learn more from giving critical feedback of peers' work than they do from giving positive feedback.

5. ACKNOWLEDGMENTS

This work is funded by the Institute of Education Sciences, under grant R305A120370.

6. REFERENCES

[1] Bardine, B. A., and Fulton, A. Analyzing the Benefits of Revision Memos during the Writing and Revision Process. *The Clearing House, 81* (4). 149–154.

[2] Calvo, R. A., Aditomo, A., Southavilay, V., and Yacef, K. The use of text and process mining techniques to study the impact of feedback on students' writing processes. *International Conference of the Learning Sciences*, 1 (2012), 416–423.

[3] Fitzgerald, J. Research on Revision in Writing. *Review of Educational Research, 57* (4). 481–506.

[4] Fitzgerald, J., and Markham, L. R. Teaching children about revision in writing. *Cognition Instruct, 4* (1). 3–24.

[5] Fitzgerald, J., and Stamm, C. Effects of Group Conferences on First Graders' Revision in Writing. *Writ Commun, 7* (1). 96–135.

[6] Flower, L., and Hayes, J. A cognitive process theory of writing. *Coll Compos Commun, 32* (4). 365–387.

[7] Gielen, S., Peeters, E., Dochy, F., Onghena, P., and Struyven, K. Improving the effectiveness of peer feedback for learning. *Lear Instr, 20* (4). 304–315.

[8] Hashemi, H. B., and Schunn, C. D. A Tool for Summarizing Students' Changes across Drafts. In *Intelligent Tutoring Systems* (Honolulu, HI, 2014), Springer International Publishing, pp. 678–682.

[9] Hillocks, G. J. The Interaction of Instruction, Teacher Comment, and Revision in Teaching the Composing Process. *Res Teach Engl, 16* (3). 261–278.

[10] Keen, J. Strategic revisions in the writing of Year 7 students in the UK. *The Curriculum Journal, 21* (3). 255–280.

[11] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the Association for Computational Linguistics: Systems Demonstrations* (2014), pp. 55–60.

[12] Morris Kindzierski, C. M. "I Like It the Way It Is!": Peer-Revision Writing Strategies for Students With Emotional and Behavioral Disorders. *Preventing School Failure: Alternative Education for Children and Youth, 54* (1). 51–59.

[13] Patchen, M. *Peer Review of Writing: Learning From Revision Using Peer Feedback and Reviewing Peers' Text*. Doctoral dissertation, University of Pittsburgh, 2011.

[14] Patthey-chavez, G. G., Matsumura, L. C., and Valdés, R. Investigating the process approach middle to writing instruction in urban schools. *Journal of Adolescent & Adult Literacy, 47* (6). 462–476.

[15] Peterson, S. Peer Response and Students' Revisions of their Narrative Writing. *Educational Studies in language and Literature, 3*. 239–272.

[16] Smede, S. D. Interior Design: Revision as Focus. *Engl J, 90* (1). 117–121.

[17] Yagelski, R. P. The role of classroom context in the revision strategies of student writers. *Res Teach of Engl, 29*, (2). 216–238.

Convergent Validity of a Student Model: Recent-Performance Factors Analysis

Ilya Goldin
Center for Digital Data, Analytics, and Adaptive Learning
Pearson
ilya.goldin@pearson.com

April Galyardt
University of Georgia
110 Carlton St.
Athens, GA
galyardt@uga.edu

ABSTRACT

Models of student performance can incorporate a skill decomposition that lists the skills that each activity requires. A good model must be sensitive to improvements in skill decomposition. We validate the Recent-Performance Factors Analysis model of student performance by checking its sensitivity to the skill decomposition. We use a dataset from a tutoring system where the skill model has been improved by the Learning Factors Analysis algorithm for skill model refinement and by expert validation. We find that R-PFA reflects improvements in the skill model, providing evidence of convergent validity of R-PFA. We argue that R-PFA may be sensible as a predictive model in Learning Factors Analysis because of its convergent validity and because the R predictor of R-PFA represents mastery-aligned learning curves.

1. INTRODUCTION

Predictive models of student performance often incorporate a skill model. For example, the Additive Factors Model [3] embeds a Q-matrix [11, 1] to relate prior practice on a skill to subsequent practice on the same skill. Bayesian Knowledge Tracing [4] similarly uses a skill model in that all BKT parameters are specific to a skill.

A skill model annotates instructional activities in terms of the skills that the activities require. This tagging can be wrong, or at least suboptimal, degrading instruction in several ways. For instance, if the tagging fails to distinguish two skills, it will treat all assessments of the two separate skills as assessments of one combined skill. In fact, because a student may have differential mastery of the two skills, the combined assessment may cause a tutoring system to call for extraneous practice for one skill, and insufficient practice for another. It follows that the refinement of a skill tagging of activities can advance instruction and assessment.

When a predictive model of student performance incorporates a skill model, we can validate the performance model by seeing if it is sensitive to changes in the skill model. A learning curve represents the "power relationship between the error rate of performance and the amount of practice" [3], plotting average error across students at every practice opportunity. If the curve treats a whole curriculum as one skill, its slope will be flat, because there will be both drops and spikes in the error rates as students learn one part of the curriculum after another. If we plot separate curves for distinct skills, their slopes will not be flat, corresponding to error rates dropping as students learn. This is the intuition for the Learning Factors Analysis algorithm [3], which searches the space of possible refinements to a skill model.

Prior study of representations of recent student performance, including box and exponential kernels with a range of bandwidths, produced the Recent-Performance Factors Analysis (R-PFA) model [6, 5]. In the recency representations with the highest predictive accuracy, the weight given to the each observation decreased with the age of the observation, placing $\sim 50\%$ of weight on the last 2 attempts, and $\sim 80\%$ on the last 5. This optimal weighting was consistent across real data and a variety of simulated student behaviors.

The current work validates R-PFA by checking whether its fit to data is improved by sensible changes to the skill tagging in a dataset. The following section describes a dataset and its multiple skill models, and presents R-PFA and several comparison models. The subsequent section reports that R-PFA and the other models are all sensitive to improved skill tagging, but R-PFA has the highest predictive accuracy among the models. Finally, we discuss how R-PFA may be interpreted as representing mastery-aligned learning curves [8], and R-PFA may fit within the Learning Factors Analysis algorithm for skill model refinement.

2. METHODS

We evaluate R-PFA on a dataset in which the skill tagging has been well-studied and revised [7], originating from Cognitive Tutor Geometry by Carnegie Learning [10, 2]. This tests R-PFA in two ways; first, how will R-PFA perform in terms of predictive accuracy? Second, does R-PFA agree with prior refinement of the skill model in this dataset [7]?

This Geometry dataset has three skill models that vary in how they treat "forward" and "backward" computations of area of geometric figures [7]. The original tagging (called Merged) separates area computation by geometric shape (square, circle, etc.), but merges together forward and backward computation. The Circle-Square tagging has separate

skills for the forward and backward computations for circles and squares. The Distinct tagging has separate forward and backward skills for each of many shapes. The geometry data set contains 38,426 unique actions by 82 students. The total number of skills in each tagging is 56 in Merged, 58 in Circle-Square, and 66 in Distinct.

We compare R-PFA to baseline models Item Response Theory 1PL, Additive Factors Model [3] and Performance Factors Analysis [9] (Eqs. 1-4). All student and skill intercepts and slopes are "random", that is, drawn from a common distribution. Treating skill parameters as random "borrows strength" for their estimation by proposing that infrequently practiced skills ought to have similar parameters as skills for which more data are available. Notation: j indexes skills, i indexes students, t indexes practice opportunities. T_{ijt} is the count of prior practice, S_{ijt} is the count of prior successes, and F_{ijt} is the count of prior failures.

$$\text{IRT 1PL} \quad \theta_i + \beta_j \tag{1}$$

$$\text{AFM} \quad \theta_i + \beta_j + \gamma_j T_{ijt} \tag{2}$$

$$\text{PFA} \quad \theta_i + \beta_j + \alpha_j S_{ijt} + \rho_j F_{ijt} \tag{3}$$

$$\text{R-PFA} \quad \theta_i + \beta_j + \delta_j R_{ijt} + \rho_j F_{ijt} \tag{4}$$

R_{ijt} is the proportion of recent successes in R-PFA (Eq. 5):

$$\text{exponential kernel } R_{ijt} = \frac{\sum_{p=-2}^{t-1} d^{(t-p)} X_{ijp}}{\sum_{p=-2}^{t-1} d^{(t-p)}} \tag{5}$$

3. RESULTS AND DISCUSSION

3.1 Predictive Accuracy

We compare predictive model accuracy in terms of AIC, a metric that rewards models for predictive accuracy and penalizes them for using excessive parameters. AIC is comparable to cross-validation with a prediction error loss function, but is more appropriate for sparse datasets, such as when only a handful of students may practice a skill [6].

Table 1: Predictive accuracy (lower AIC is better).

	Skill tagging		
Model	Merged	Cir-Sq	Distinct
IRT 1PL	21652	21538	21523
AFM	21373	21252	21272
PFA	21326	21197	21211
exp R-PFA r(0.7), f(0.1)	21142	20969	21003
exp R-PFA "best" from search	21134	20949	20977
"best" decay rates: R, F	0.7, 0.3	0.5, 0.3	0.4, 0.3

For all 3 skill taggings, R-PFA has higher predictive accuracy than the other models, with PFA, AFM, and Item Response Theory 1PL following in that order (Table 1). IRT 1PL has the lowest predictive accuracy, likely because it does not reflect learning over time. At the best-performing R and F decay weights from prior work (0.7 and 0.1, respectively), the number of parameters in PFA and R-PFA is exactly the same, and R-PFA's advantage in AIC over PFA is due to increased predictive accuracy.

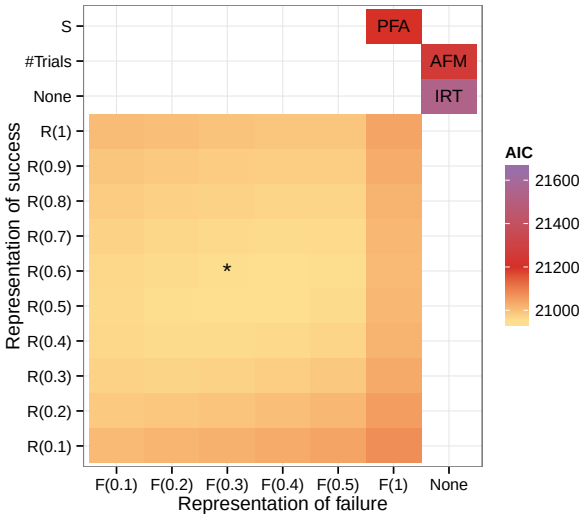


Figure 1: AIC for all 63 models on Circle-Square tagging. * denotes the best overall model.

Searching over decay rates shows that R-PFA is robust to a range of rates (Fig. 1). Even though the strictly lowest AIC uses decay rates that differ from prior work, this effect is smaller (26 points on Distinct, Table 1) than the effect of using R-PFA over other models or of improving the skill tagging, and R-PFA's performance degrades gracefully. Tuning decay rates separately for skill models has only a marginal benefit, and may confound skill model comparison.

We compare the learning curves of the 4 performance models (Fig. 2 and 3), omitting practice opportunities with fewer than 5 students. The red curves show the empirical percent correct at each opportunity, with a binomial 95% Bayesian credible interval that uses a Jeffreys prior. For example, at the 1st opportunity for circle-area backward, the mean is 45% correct, with CI (21%, 41%). The intervals make no adjustment for multiple comparisons (at each practice opportunity), so they are overly narrow, but remain useful for comparing model predictions to student performance.

The model fit curves (black) show the 2.5th and 97.5th quantiles of the model predictions. A model should predict that some students have a lower probability of a correct answer than the population percent correct, and other students, respectively, have a higher probability. If a model fits the data well, the black model curves should be centered over the empirical red curves, but should have wider bars on early attempts where there are many students in the sample.

R-PFA consistently tracks the empirical learning curve more closely than the alternative models for all 6 skills, but most clearly in circle-area backward and square-area backward (Fig. 2). Consider AFM and R-PFA predictions on circle-area backward opportunity 1: AFM predicts that 60% of students will respond correctly, when only 45% do; in fact 95% of model predictions for AFM are above the empirical percent correct. AFM produces many false positives on this early opportunity. For R-PFA, the model predictions are

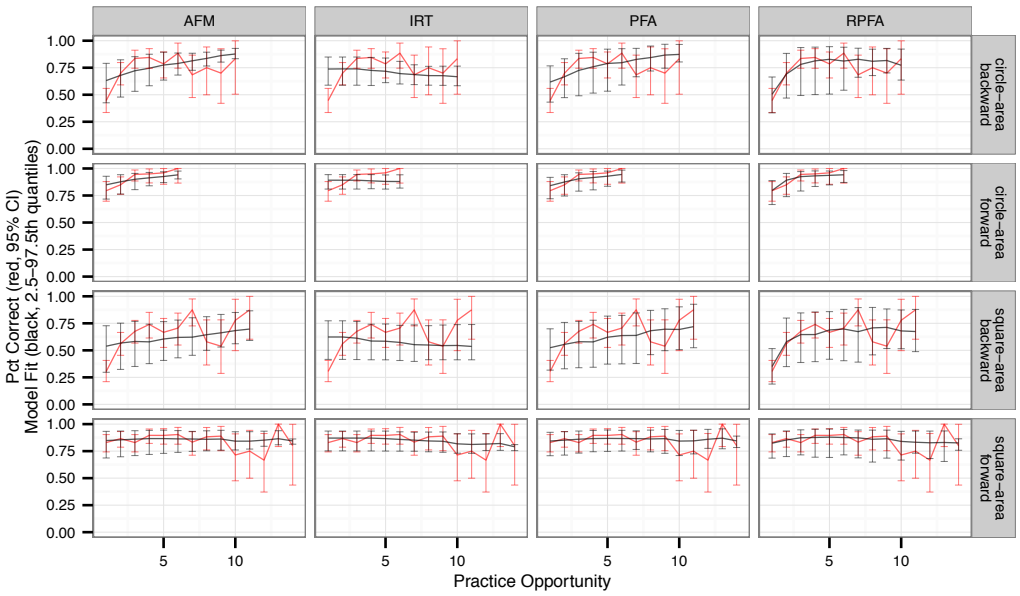


Figure 2: Empirical learning curves (red) and model fits (black) for newly split skills tagged in Cir-Sq.

centered over the empirical percent correct, and producing fewer false positives. On opportunity 4, AFM predictions are too low. AFM underestimates the amount of learning that has occurred, while R-PFA predictions track the empirical percent correct. Moreover, the R-PFA predictions range from below 0.5 to above 0.9, indicating that R-PFA is able to distinguish students who have learned the skill from those who need more practice.

3.2 Sensitivity to Skill Tagging

All models except IRT 1PL (which has the worst AIC) replicate the ranking of the three skill taggings [7]. The Cir-Sq tagging provides the best balance of predictive accuracy and data fit, compared to the Distinct tagging (which may be more granular than necessary to describe this dataset), and the Merged tagging (not sufficiently granular). While both the tagging and R-PFA are merely imperfect models, the replication provides convergent evidence for the validity of both. Skill model refinement need not improve predictive accuracy, but if it does and if the refinement makes sense in terms of instruction and cognition, that provides some evidence that the change represents an aspect of learning that is reflected in student performance.

R-PFA with the Merged tagging has a lower AIC score than any other model with the Cir-Sq tagging. Even though the Cir-Sq split is sensible and R-PFA benefits from it, R-PFA is more robust to the absence of such a split than other models. This shows in R-PFA's fit to the learning curve of circle-area (Fig. 3). AFM's predictions do not reflect the performance drop on opportunities 11 and later, but R-PFA does. This decrease motivated splitting circle-area into forward and backward skills, as in Cir-Sq [7], but R-PFA hews to the curve even without the split.

3.3 R-PFA Disaggregates Learning Curves

R-PFA effectively disaggregates the learning curves of individual students. Traditional learning curves are aligned at the first practice opportunity. Mastery-aligned curves [8] are aligned in terms of the opportunity at which students first achieve mastery. Traditional curves may conceal learning, such as if students differ in their relevant skill knowledge before their first observed practice opportunity, or if a skill model conflates two distinct skills [8]. The proportion of recent successes R by itself is a decay-weighted moving average that represents (in a non-parametric, non-model based way) the probability of mastery. R reflects the mastery-aligned curve in a predictive model, analogous to how total practice T represents the traditional learning curve in AFM.

The slope of R in R-PFA requires a different interpretation than the slope of T. A history of practice where recent success is positively associated with subsequent success (and recent failure is positively associated with subsequent failure) will have a positive slope, i.e., a positive effect on predicting the outcome. Practice relatively far in the past, whether successful or not, will have comparatively little effect on the prediction. (With the decay rate $d = 0.7$, practice older than about 5 opportunities has little effect on the prediction [6].)

One case in which the direction of the slopes of R and T may differ is in the case of a "blip" [4], i.e., when two skills follow each other in one curve, and the success rate drops in the middle of the curve, corresponding to the beginning of practice on a second skill (circle-area in Fig. 3). The slope of T ought to be flat in such a circumstance, which has been taken to mean that the skill may require a split. The slope of R will be positive, representing the fact that there is learning along the first disaggregated curve, and then along the second disaggregated curve. In fact, slopes of circle-area

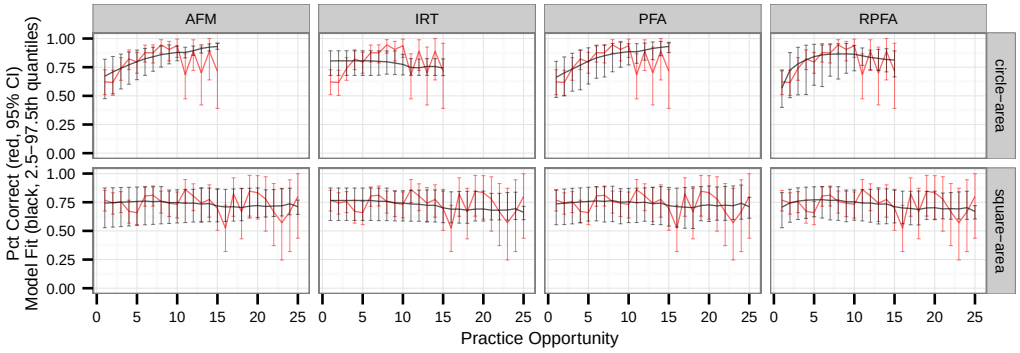


Figure 3: Learning curves (red) and model fits (black) for skills tagged in Merged, but later split in Cir-Sq.

are positive according to both AFM and R-PFA, and slopes of square-area are flat according to both AFM and R-PFA, suggesting that the slope of T is not an ideal heuristic for choosing a skill for a split.

An alternative heuristic is that when the slope of R is negative or flat, that implies that even disaggregated, mastery-aligned learning curves are a poor representation of the skill at hand. This suggests issues with the tagging of problems for this skill. This is a reasonable opportunity to invite experts to investigate "difficulty factors" for this skill, and to use LFA to apply these factors.

4. CONCLUSIONS

This investigation validates the R-PFA model of student performance in predictive accuracy on a real-world dataset. It provides convergent validity evidence for R-PFA by showing that it is sensitive to changes in a well-documented skill tagging, and yet robust to noise in a skill model. Given that no skill model is perfect, a predictive model that is accurate even in the face of such noise could be an asset to adaptive learning technologies.

The skill tagging refinement algorithm LFA [3], which incorporates AFM, may benefit by switching to R-PFA. LFA uses AFM in two ways: as a component in A* search, and as an interpretable learning curve slope. R-PFA may be a better component in A* search, because it is a more accurate model that is still sensitive to skill model changes, and because it reflects a mastery-aligned curve rather than an aggregate curve. The interpretation of the slope parameter is different, but sensible.

5. ACKNOWLEDGMENTS

We thank Ran Liu for thoughtful comments and assistance.

6. REFERENCES

[1] T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence Educational Data Mining Workshop*, 2005.

[2] M. Bernacki and S. Ritter. Motivation for learning HS Geometry 2012 (geo-pa). Dataset 748 in DataShop. Retrieved from `https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=748`, 2014.

[3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of 8th International Conference on Intelligent Tutoring Systems*, pages 164–175, 2006.

[4] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[5] A. Galyardt and I. Goldin. Recent-Performance Factors Analysis. In J. Stamper, Z. Pardos, M. Mavrikis, and B. McLaren, editors, *Proceedings of 7th International Conference on Educational Data Mining*, pages 411–412, 2014. (Poster paper).

[6] A. Galyardt and I. M. Goldin. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining*, Accepted.

[7] R. Liu, K. Koedinger, and E. McLaughlin. Interpreting model discovery and testing generalization to a new dataset. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.

[8] R. C. Murray, S. Ritter, T. Nixon, and al. Revealing the learning in learning curves. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Proceedings of 16th International Conference on Artificial Intelligence in Education*, pages 473–482, 2013.

[9] P. I. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–a new alternative to Knowledge Tracing. In *Proceedings of 14th International Conference on Artificial Intelligence in Education*, pages 531–538. IOS Press, 2009.

[10] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. T. Corbett. The Cognitive Tutor: Applied research in mathematics education. *Psychonomics Bulletin & Review*, 14(2):249–255, 2007.

[11] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

POSTER AND DEMO PAPERS

Automatic Grading of Short Answers for MOOC via Semi-supervised Document Clustering

Shumin Jing
Warner School of Education
University of Rochester, Rochester, NY, USA
jingshumin@gmail.com

ABSTRACT

Developing an effective and impartial grading system for short answers is a challenging problem in educational measurement and assessment, due to the diversity of answers and the subjectivity of graders. In this paper, we design an automatic grading approach for short answers, based on the non-negative semi-supervised document clustering method. After assigning several answer keys, our approach is able to group the large amount of short answers into multiple sets, and output the score for each answer automatically. In this manner, the effort of teachers can be greatly reduced. Moreover, our approach allows the interaction with teachers, and therefore the system performance could be further enhanced. Experimental results on two datasets demonstrate the effectiveness of our approach.

Keywords

Clustering, semi-supervised learning, short-answer grading

1. INTRODUCTION

Grading short answers is a challenging problem in the conventional educational measurement and assessment [6, 4], due to the diversity of answers and the subjectivity of graders. Especially, in the era of the massive open online course (MOOC), this problem becomes critical. MOOC provides plenty of courses, and has attracted over 10 million users during the past few years. However, traditional assessments are not suitable for MOOC. For example, in most MOOC platforms, short answers appear frequently in various quizzes and exams. Obviously, hiring lots of graders is not a feasible solution. Thus, it is very necessary to develop an automatic grading system for short answers. The automatic grading system for short-answers has been widely studied during the past decade [2]. Most recently, a system named "Powergrading" was presented by Microsoft Research, which achieved quite impressive performance [1].

We would argue that clustering is a straightforward solution to automatic grading. For short answer grading, the motivation of using clustering is that, the similar short answers should have high similarity values, while the dissimilar ones should have low similarity values. Therefore, those similar short answers could be assigned into the same group. We can then infer the final scores of those answers according which groups they belong to.

In this paper, we aim to design an automatic grading approach for short answers. Our approach is expected to solve the assessment challenge in MOOC. Moreover, it can also be applied to traditional educational assessment scenario, to reduce the efforts of teachers. We will present the methodology of our approach, discuss its influence in online education, and report the quantitative results and analysis.

2. METHODOLOGY

2.1 Feature Representation

In our problem, each short answer can be treated as a short document. Let $W = \{f_1, f_2, \cdots, f_m\}$ denote a complete vocabulary set of the short answers after the stopwords removal and words stemming operations. We can get the term-frequency vector X_i of short answer d_i as follows

$$X_i = [x_{1i}, x_{2i}, \cdots, x_{mi}]^\top \tag{1}$$

$$x_{ji} = t_{ji} \times \log(\frac{n}{\text{idf}_i}) \tag{2}$$

where t_{ji}, idf_i, n denote the term frequency of word f_j in short answer d_i , the number of short answers containing word f_j, and the total number of documents in the corpus, respectively.

By using X_i as a column, we can construct the term-short-answer matrix X.

2.2 Semi-Supervised Clustering for Short-answer Grading

We observe that, the label information of short answers is neglected in the basic document clustering approach. However, by leveraging the expertise of teachers, we can usually get some useful information. For example, teachers will tell us which two answers are essentially similar to each other, although they look quite different on the first sight.

To make use of such useful information, we propose a semi-supervised document clustering approach. The basic idea

is to add some constraints, including positive ones and negative ones. The former one shows us which short answers are similar, and we can always put them into the same cluster. On the other hand, the latter one tells us which short answers cannot be grouped together.

Inspired by the semi-supervised clustering algorithm [3, 5], we present the non-negative semi-supervised document clustering (SSDC) algorithm for short-answer grading as follows.

Let $A = X^\top X$ denote the document (e.g., short-answer) similarity matrix. In our approach, we first employ the symmetric non-negative tri-factorization as follows

$$A = QSQ^\top \tag{3}$$

where Q is the cluster indicator matrix. Each element in Q represents the degree of association of the short-answer d_i with cluster j. The cluster membership information is determined by seeking an optimization matrix S.

In the semi-supervised setting, we are given two sets of pairwise constraints on the short-answers, including the must-link constraints C_{ML} and cannot-link constraints C_{CL}. Every pair in C_{ML} means this pair of short-answers should belong to the same cluster; every pair in C_{CL} means this pair of short-answers should belong to different clusters.

Then, the objective function of SSDC algorithm is

$$\begin{aligned} &J = \arg\min \|\bar{A} - QSQ^\top\|^2 \\ &s.t., S \geq 0, Q \geq 0, \end{aligned} \tag{4}$$

where $\bar{A} = A - R_+ + R_-$. R_+ and R_- are two penalty matrices, considering the two constraint sets C_{CL} and C_{ML}.

The problem (4) can be solved efficiently using the standard gradient descent algorithm. The update rules of S and Q are given below

$$S_{ij} = S_{ij} \frac{(Q^\top \bar{A} Q)_{ij}}{(Q^\top QSQ^\top Q)_{ij}} \tag{5}$$

$$Q_{ij} = Q_{ij} \frac{(\bar{A}QS)_{ij}}{(QSQ^\top QS)_{ij}}. \tag{6}$$

After obtaining the optimized S and Q, we can use them to infer the cluster labels for each short answer.

Finally, we can assign the score for each short-answer. For example, we know that the score of one template answer is 8.0. If another short-answer and this template answer belong to the same cluster, then the score of this short-answer should be close to 8.0. We also design a weighting strategy to adjust this score, based on the distance to the template answer.

3. EXPERIMENTS

We utilize the data set provided by Microsoft Research, which is also analyzed in the paper (Basu, Jacobs & Vanderwende, 2013). It contains the responses from 100 and 698 crowdsourced workers to each of 20 short-answer questions. These questions are taken from the 100 questions published by the United States Citizenship and Immigration Services as preparation for the citizenship test. It also contains labels of response correctness (grades) from three judges for a subset of 10 questions for the set of 698 responses (3 x 6980 labels).

Table 1: The Results on MSR Dataset and MOOC Dataset.

Method	MSR Dataset	MOOC Dataset
DC	85.2%	74.1%
Semi-supervised DC	87.5%	78.9%

Besides, we also collect some short answers from MOOC websites. We will evaluate the performance of our approach on both datasets.

We evaluate the performance of our approach on the MSR dataset and MOOC dataset. As we have the ground truth information, we can report the accuracy of clustering algorithms. Table 1 shows the accuracies of our approach and the baseline method DC under different settings. It shows that our semi-supervised document clustering method always achieves better performance than DC on two datasets.

4. CONCLUSIONS AND FUTURE WORK

We studied the educational assessment problem in MOOC. In this paper, we proposed an automatic grading approach for short answers. By leveraging the benefits of document clustering, our approach was able to assign a large amount of short answers into different groups, and infer their scores accordingly. Moreover, we designed a semi-supervised approach, which is able to incorporate the expertise of teachers. The proposed approach fits the requirements of MOOC. Results on two datasets showed the effectiveness of our approach. Our paper provides an effective solution to the educational assessment problem. In the future, we will design more computer-aided systems to address the educational assessment problem.

5. REFERENCES

[1] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *TACL*, 1:391–402, 2013.

[2] C. Brew and C. Leacock. Automated short answer scoring. *Handbook of automated essay evaluation: Current applications and new directions*, (136), 2013.

[3] Y. Chen, M. Rege, M. Dong, and J. Hua. Non-negative matrix factorization for semi-supervised data clustering. *Knowl. Inf. Syst.*, 17(3):355–379, 2008.

[4] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppala. Review of recent systems for automatic assessment of programming assignments. *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, 2010.

[5] L. Lelis and J. Sander. Semi-supervised density-based clustering. In *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*, pages 842–847, 2009.

[6] C. R. Reynolds, R. B. Livingston, V. L. Willson, and V. Willson. Measurement and assessment in education. *Pearson Education International*, 2010.

Discovering students' navigation paths in Moodle

Alejandro Bogarín
Department of Computer Science
University of Cordoba, Spain
(+34) 679 30 54 86
abogarin@uco.es

Cristóbal Romero
Department of Computer Science
University of Cordoba, Spain
(+34) 653 46 28 13
cromero@uco.es

Rebeca Cerezo
Department of Psychology
University of Oviedo, Spain
(+34) 627 60 70 21
cerezorebeca@uniovi.es

ABSTRACT
In this paper, we apply clustering and process mining techniques to discover students' navigation paths or trails in Moodle. We use data from 84 undergraduate Psychology students who followed an online course. Firstly, we group students using on Moodle's usage data and the students' final grades obtained in the course. Then, we apply process mining with each cluster/group of students separately in order to obtain more specific and accurate trails than using all logs together.

Keywords
Clustering, process mining, navigation paths, trails in education.

1. INTRODUCTION
One of the current promising techniques in EDM (Educational Data Mining) is Educational Process Mining (EPM). The main goal of EPM is to extract knowledge from event logs recorded by an educational system [4]. It have been observed that students show difficulties when learn in hypermedia and Computer Based Learning Environments (CBLEs) due to these environments seems to be highly cognitive and metacognitive demanding [1]. In this sense, the models discovered by EPM could be used: to get a better understanding of the underlying educational processes, to early detect learning difficulties and generate recommendations to students, to help students with specific learning disabilities, to provide feedback to either students, teachers or researchers, to improve management of learning objects, etc. In a previous work [2], we found two problems when using EPM: 1) the model obtained could not fit well to the general students' behaviour and 2) the model obtained could be too large and complex to be useful for a student or teacher. In order to solve these problems, we proposed to use clustering to improve both the fitness and comprehensibility of the obtained models by EPM. However, in this paper we propose to use a Hypertext Probabilistic Grammar (HPG) algorithm instead of Heuristics Net [2] because it provides more informative graphs.

2. METHODOLOGY
A traditional approach would use all event log data to reveal a process model of student's behaviour . Nevertheless, in this paper, we propose an approach that uses clustering for improving EPM (see Figure 1). The proposed approach firstly applies clustering in order to group students with similar features. And then, it applies process mining for discovering more accurate models of students' navigation paths or trails. In fact, we propose to use two different grouping methods:

1) Clustering students directly by using the students' grades obtained in the final exam of the course.
2) Clustering students by using a clustering algorithm over the student's interaction with the Moodle's course.

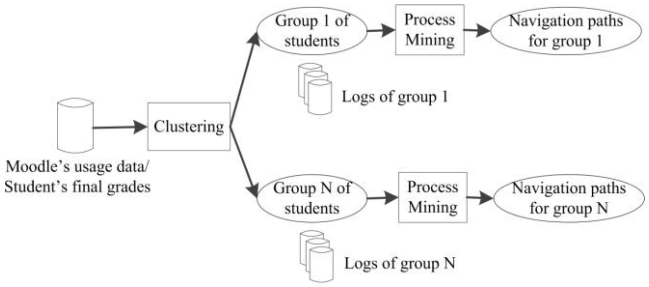


Figure 1: Proposed approach for discovering students' navigation paths.

3. DESCRIPTION OF THE DATA AND EXPERIMENTS
In this work we have used real data collected from 84 undergraduate Psychology students who followed a Moodle course. Firstly, we have divided the student's log provided by Moodle in two different ways. In a first way, we divided directly the original log file into two datasets: one that contains the 68 students who passed the course and other with the 16 students who failed. In the second way, we have used the Expectation-Maximization (EM) clustering algorithm provided by Weka [6] in order to group together students of similar behaviour when using Moodle. In this case we have obtained three clusters/datasets with the following distribution:

- **Cluster 0**: 23 students (22 pass and 1 fail).
- **Cluster 1:** 41 students (39 pass and 2 fail).
- **Cluster 2**: 20 students (13 fail and 7 pass).

After clustering, we applied EPM through HPG over the previous datasets. We have used the HPG model in order to efficiently mine trails or navigation paths [3]. HPG uses a one-to-one mapping between the sets of non-terminal and terminal symbols. Each non-terminal symbol corresponds to a link between Web pages. Moreover, there are two additional artificial states, called *S* and *F*, which represent the start and finish states of the navigation sessions respectively. The probability of a grammar string is given by the product of the probability of the productions used in its derivation. The number of times a page was requested, and the number of times it was the first and the last page (state) in a session, can easily be obtained from the collection of student navigation sessions. The number of times a sequence of two pages appears in the sessions gives the number of times the corresponding link was traversed. The aim is to identify the subset of these trails that correspond to the rules that best characterize the student's behavior when visiting the Moodle course. A trail is included only if its derivation probability is above a cut-point.

The cut-point is composed of two distinct thresholds (support and confidentiality). The support (Sup) value is for pruning out the strings whose first derivation step has low probability, corresponding to a subset of the hypertext system rarely visited. The confidence (Con) value is used to prune out strings whose derivation contains transitive productions with small probabilities. Support and confidence thresholds give the user control over the quantity and quality of the obtained trails, while α (Alp) modifies the weight of the first node in a student navigation session: when α is near 0, only those routes that start in a node which started a session are generated; when α is near 1, all weights are completely independent of the order within the session.

4. RESULTS

We have carried out several experiments with the HPG algorithm to test several configurations of number of Nodes, Links, Routes, and average route length (Avg). Results obtained when using different datasets and parameters are displayed in Table 1.

Table 1. Results with different datasets and configurations.

Dataset	Alp	Sup	Con	Nodes	Links	Routes	Avg
Fail	0,2	0,05	0,5	8	7	12	3,85
Pass	0,2	0,05	0,5	12	11	20	3,81
Cluster0	0,2	0,05	0,5	8	6	12	4,16
Cluster1	0,2	0,05	0,5	9	7	14	4,14
Cluster2	0,2	0,05	0,5	5	4	6	2,75
Fail	0,4	0,06	0,3	15	15	27	3,8
Pass	0,4	0,06	0,3	25	27	47	3,96
Cluster0	0,4	0,06	0,3	13	12	21	3,66
Cluster1	0,4	0,06	0,3	15	17	29	4,11
Cluster2	0,4	0,06	0,3	12	9	18	3,66
Fail	0,5	0,06	0,3	20	19	36	4
Pass	0,5	0,06	0,3	37	41	72	4,07
Cluster0	0,5	0,06	0,3	19	17	31	3,7
Cluster1	0,5	0,06	0,3	20	21	38	4,19
Cluster2	0,5	0,06	0,3	12	9	18	3,66

Table 1 show that the smaller and more comprehensible models were obtained using logs from students who failed (Fail dataset) and students of Cluster 2. On the other hand, the models obtained with the other datasets were much bigger and complex. We think that this may be due to:

- Both dataset Fail and Cluster 2 contain mainly information about bad students who failed the course. This type of students has a low interaction with Moodle and so, they show only some frequent navigation paths.
- Datasets Pass, Cluster0 and Cluster1 contain mainly information about good students who pass the course. This type of students has a high interaction with Moodle and so, they show more frequent navigation paths.

Finally, we show an example of obtained model when using the Cluster2 dataset. In Figure 2, each node represents a Moodle's Web page, and the directed edges (arrows) indicate how the students have moved between them. These paths can be stochastically modeled as Markov chains [5] on the graph, where the probability of moving from one node to another is determined by which Web page the student is currently visiting. Edge thickness varies according to edge weight; this allows the learning designer to quickly focus on the most important edges, ignoring those that have very low weights. In addition, line widths and numerical weights are also available.

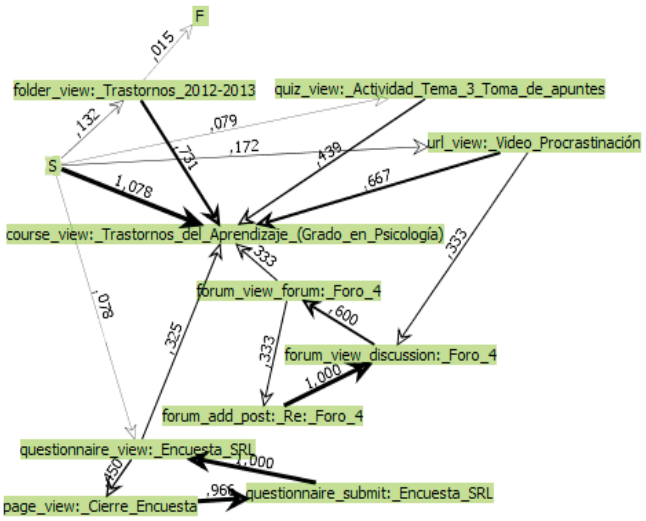


Figure 2: Navigation paths of Cluster 2 students.

Starting from Figure 2 we can see and detect what are the most frequent actions (view forum X, view questionnaire Y, view quiz Z, etc.) and in which order (navigation paths or trails) were done/followed by Cluster 2 students (normally fail students).

5. REFERENCES

[1] Azevedo, R., Behnagh, R., Duffy, M., Harley, J. M., & Trevors G. J. Metacognition and self-regulated learning in student-centered learning environments. *Theoretical foundations of student-center learning environments*. Erlbaum, Mahwah, NJ, 2nd edition, 216–260, 2012.

[2] Bogarin, A. Romero, C., Cerezo, R., Sanchez, M. Clustering for improving Educational Process Mining. *Learning Analytics and Knowledge Conference*, Indianapolis, 11-14.

[3] Borges, J., Levene, M. Data Mining of user navigation patterns. Proc. of Workshop Web Usage Analysis and User Profiling. San Diego, 2000. pp. 31-36.

[4] Pechenizkiy, M., Trcka, N., Vasilyeva, E., van der Aalst, W.M., & De Bra, P. 2009. Process Mining Online Assessment. Data. *Educational Dada Mining Conference*, Cordoba, Spain, 279-288.

[5] Kemeny, J.G., Snell, J.L. Finite Markov chains. Princeton: Van Nostrand. 1960

[6] Witten, I.H., Eibe, F., Hall, M.A. Data Mining, Practical Machine Learning Tools and Techniques. Third Edition. Morgan Kaufman Publishers, 2001.

Teacher-Student Classroom Interactions: A Computational Approach

Arnon Hershkovitz
School of Education
Tel Aviv University
Tel Aviv, ISRAEL
arnonhe@tauex.tau.ac.il

Agathe Merceron
Media Informatics department
Beuth University of Applied Sciences
Berlin, GERMANY
merceron@beuth-hochschule.de

Amran Shamaly
School of Education
Tel Aviv University
Tel Aviv, ISRAEL
amranshamaly@mail.tau.ac.il

ABSTRACT
Teacher-student interactions are key to most school-taught lessons. We present a new approach to studying these interactions; this approach is based on a fine-grained data collection, using quantitative field observations (QFOs), which relies on a well-established theoretical framework. The data collected can be analyzed in various methods to address different types of research questions; we give some examples to demonstrate this potential.

Keywords
Teacher-student interactions, quantitative field observations, different analysis approaches.

1. INTRODUCTION
Since the early days of Plato, over 2,300 years ago, dialogues were at the heart of the teaching practice. For as long as classroom teaching exists, teacher-student interactions have been the key to most school-taught lessons, hence studying these interactions is decades-old. Many studies in this field that have used classroom observations, often manually documented each occurrence of a teacher-student interaction, usually by observing a small cohort of students at a time or by observing individual students based on an interval-based protocol (e.g., Good & Brophy, 1970; Cameron, Cook & Tankersley, 2012; Luckner & Pianta, 2011). We use a digital data collection tool—a tablet app developed specifically for this purpose—in order to conduct quantitative field observations (QFOs). Although documenting each occurrence of a teacher-student interaction, data is not collected at the student-level (i.e., students are not labeled, only interactions), which makes it feasible to have a single person observing a whole class and still document every interaction during it. Once the class is over, the data is ready to be analyzed. More than that, this fine-grained data is time-stamped, which allows for advanced, including temporal, analyses.

2. THEORETICAL FRAMEWORK
Good and Brophy's (1970) method, developed in the context of mathematics education, was probably the first to refer to a single student—as opposed to the whole class—while recording public classroom interactions, hence focusing on dyadic teacher-student interactions**. T**his protocol was later modified by Reyes and Fennema (1981), who considered non-public teacher-student interactions too.

These validated protocols have been in use to study various variables at different grade levels and in many learning settings. Due to their validity, fine granularity and popularity, we find these protocols very suitable for our research. Adapting and extending the original protocols to better fit to our research setting—mainly to the whole class being observed at all times—we categorize each teacher-student interaction to one of the categories described in the next sub-sections.

2.1 Response Opportunity
A response opportunity is a public attempt by an individual student or a group of students to deal with a question posed by the teacher. Interactions that fall under this category take one of four possible values: **Direct** – the teacher asks a direct question of an individual student; **Volunteer** – the teacher asks a question, waits for the students to raise their hands, then calls on one of the children who has his hand up; **Call Out Single** – the teacher asks a question and a student calls out an answer without waiting for permission to respond; **Call Out 2+** – the teacher asks a question and more than one student call out an answer without waiting for permission to respond.

2.2 Immediate Contact
An immediate interaction is a public, content-related interaction initiated by the teacher, a student or a group of students that is not preceded by a teacher's question. This category again has four values based on the interaction initiator and the number of students involved in it: **Teacher to Single**, **Teacher to 2+**, **Single to Teacher**, **2+ to Teacher**.

2.3 Behavioral Contact
These are public, behavior-related comments of the teacher. Here too, four values are defined, based on the type of behavior commented and on the targeted audience: **Discipline to Single**, **Discipline to 2+**, **Appraisal to Single**, **Appraisal to 2+**.

2.4 Procedural Contacts
These interactions are public, non-content related; they are related to students' management or to the class management, e.g., permission, supplies, or equipment. Like *Immediate*, we distinguish the interaction initiator and the number of students involved, hence its four values are: **Teacher to Single**, **Teacher to 2+**, **Single to Teacher**, **2+ to Teacher**.

2.5 Non-Public Interactions
Non-public interactions are held privately between the teacher and one or more students. As such, we assume not being able to categorize them, therefore we only code whether they were **Teacher-Afforded** or **Student-Initiated**.

3. DATA COLLECTION APP
As mentioned above, a dedicated data collection app was developed for the purpose of this study. The app, Q-TSI

(Quantifying Teacher-Student Interactions), is available for free via Google Play Store[1]. Besides coding the interaction categories, the app allows documenting the following contextual variables:

- **Learning Configuration** (whole class discussion, group work, pair work, individual work);
- **Technologies in Use by the Teacher** (blackboard, projector, smart board, book – any combination of these are allowed);
- **Technologies in Use by the Students** (book, computer, book and computer);
- **Teacher Location** –on a 4x4 division of the classroom.

Furthermore, the app allows the user to enter any (time-stamped) comment s/he finds useful. These comments might be useful to interpret the results of analyses. The data is stored locally on the observer device as a CSV file.

4. ANALYSES APPROACHES

The collected data can be analyzed in various ways in order to address a wide range of research questions. We now describe a few potential research directions we are currently considering (some will be demonstrated in the poster).

4.1 Visualization

Visualization can be a powerful tool to have an overall understanding of the classroom dynamics. Teachers can gain awareness and reflect upon their interactions with their students during the class. A typical visualization may include a time-ordered representation of the interactions, differentiated by type (e.g., by color, marker), along with values of the contextual variables. Such visualizations may assist in initially having an overview of the kinds of interactions that happen, exploring differences within classes, based on, e.g., learning configuration or technologies in use, or between classes, based on, e.g., teacher, school, grade-level, subject matter, time of day, etc.

4.2 Statistics

Basic statistics may shed light on the overall distribution of the different types of interactions in a lesson, as well as on differences within and between classes (based on variables as such as were mentioned in *4.1 Visualization*).

4.3 Time-based Patterns

Association rules, time series, statistical discourse analysis and epistemic network analysis may assist in understanding whether there are specific interactions that often occur jointly or in connection, possibly, in some specific order or in a specific context, and how occurrences of interactions evolve over time. Time in our context has at least two levels of granularity: the lesson granularity (i.e., what happens during one lesson) and the school year granularity (i.e., what happens in lessons over the weeks).

4.4 Cluster Analysis

Clustering techniques can be used to explore whether classes can be classified according to typical patterns of interactions. Several ways of describing a lesson and, consequently, of comparing lessons, can be investigated. For instance, a mere quantitative analysis can be used to characterize a lesson, that is, counting interactions and using the Euclidean distance (or alike) for clustering. A lesson can also be described as a sequence of different interactions over time, then using the Levenshtein distance (or alike) for clustering. It might be necessary to define several abstraction levels for the interactions.

4.5 Prediction

It might be possible to predict different types of interactions based on historical data, or based on contextual variables. A possible prediction might look like: "three <Response opportunity: Call out 2+> interactions and two <Procedural: Teacher to single> interactions are followed by a <Discipline: Single> interaction in 85% of the instances." Several techniques will be considered to investigate this kind of patterns, in particular, classification techniques enriched with time series and statistical discourse analysis.

4.6 Collecting More Data

In the future, additional data will be collected, such as students' log files, performance, meta-cognitive and affective measures, in order to enrich the data with more layers. These layers will allow, in turn, to ask even more questions about the data and to better investigate the role of teacher-student interactions in the learning/teaching process.

5. ACKNOWLEDGMENTS

This research is partially funded by the European Commission's Marie Curie Career Integration Grant (CIG) 618511/ARTIAC.

6. REFERENCES

[1] Brophy, J.E. & Good, T.L. 1969. Teacher-child dyadic interaction: A manual for coding classroom behavior (Report Series No. 27). Austin, TX: Texas University.

[2] Cameron, D.L., Cook, B.G., & Tankersley, M. 2012. An analysis of the different patterns of 1:1 interactions between educational professionals and their students with varying abilities in inclusive classrooms. *International Journal of Inclusive Education, 16*, 12, 1335-1354.

[3] Good, T.L. & Brophy, J.E. (1970). Teacher-child dyadic interactions: A new method of classroom observation. *Journal of School Psychology, 8*(2), 131-138.

[4] Luckner, A.E. & Pianta, R.C. 2011. Teacher–student interactions in fifth grade classrooms: Relations with children's peer behavior. *Journal of Applied Developmental Psychology, 32*, 5, 257-266.

[5] Reyes, L. & Fennema, E. (1981). *Classroom Processes Observer Manual*. Madison, WI: Wisconsin Center for Education Research.

[1] https://play.google.com/store/apps/details?id=com.gil.q_tsi

Modeling Student Learning: Binary or Continuous Skill?

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

Student learning is usually modeled by one of two main approaches: using binary skill, with Bayesian Knowledge Tracing being the standard model, or using continuous skill, with models based on logistic function (e.g., Performance Factor Analysis). We use simulated data to analyze relations between these two approaches in the basic setting of student learning of a single skill. The analysis shows that although different models often provide very similar predictions, they differ in the impact on student practice and in the meaningfulness of parameter values.

Keywords

student modeling; learning; Bayesian Knowledge Tracing; simulated data

1. INTRODUCTION

In this work we focus on modeling of student learning in the basic setting: we assume that for each student we have a sequence of answers related to a single skill and we consider only correctness of these answers, i.e., we do not take into account additional information like response times or partial correctness due to the use of hints. We work only with basic models and focus on experiments with simulated data. This setting is of course a coarse simplification, since in a real application we typically have some additional information on student answers, questions are related to multiple skills, and model extensions are used. But in order to successfully use complex models, it is necessary to have deep understanding of the base case and this understanding is still lacking. There are many feasible modeling approaches, but they are usually proposed and studied independently and their relations, similarities, and differences have not been well studied. The use of simulated data allows us to analyze behaviour of models in detail thanks to the knowledge of "ground truth" values; moreover, we can manipulate in controlled way generation of data and thus easily evaluate behaviour of models under different assumptions.

2. MODELING STUDENT LEARNING

Most approaches to modeling of student learning can be viewed as hidden Markov models (also called latent process models, state-space models). We assume a hidden (latent) state variable (called "skill") and two types of equations. Observation equation describes the dependence of observed variables (correctness of answers) on the hidden variable (skill). State equation describes the change of the hidden variable (i.e., learning). There are two main types of models depending on whether the latent skill is binary or continuous. It is in principle possible to consider discrete skill with more than two states, but such models are not commonly used. The standard form of a binary skill model is Bayesian Knowledge Tracing (BKT) [1]. Models based on continuous latent skill typically use logistic function for observation equation, they differ in their approach to skill estimation.

Bayesian Knowledge Tracing assumes a sudden change in knowledge. It is a hidden Markov model where skill is a binary latent variable (either learned or unlearned). Figure 1 illustrates the model; the illustration is done in a non-standard way to stress the relation of the model to the model with continuous skill. The estimated skill is updated using a Bayes rule based on the observed answers; the prediction of student response is then done based on the estimated skill. Note that although the model is based on the assumption of binary skill, the skill estimate is actually continuous number (in the $[0, 1]$ interval).

Models which utilize the assumption of continuous latent skill consider skill in the $(-\infty, \infty)$ interval and for the relation between the skill and the probability of correct answer use the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$. Although it would be possible to consider also other functions, the logistic function is currently the standard choice. As a simple model of learning we consider a simple linear growth of the skill (Figure 1). More specifically, for the initial skill θ_0 we assume normally distributed skill $\theta_0 \sim N(\mu, \sigma^2)$ and for the change in learning we consider linear learning $\theta_k = \theta_0 + k \cdot \Delta$, where Δ is either a global parameter or individualized learning parameter (in that case we assume a normal distribution of its values). This model is a simplified version of the Additive Factors Model [3]; the original additive factor model uses multiple skills. A principled way of estimating continuous skills is the Bayesian approach, which computes not just a point estimate of skill, but a distribution over skill capturing also the uncertainty of the estimate. This approach be implemented for example using particle filter, i.e., discretized

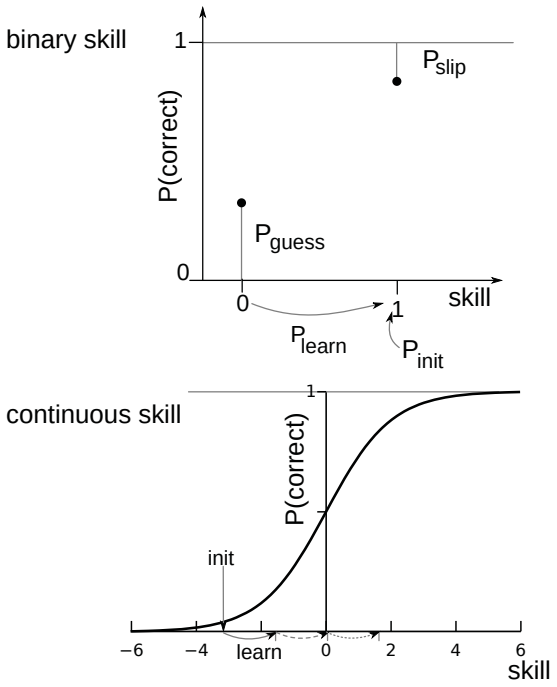


Figure 1: Binary and continuous skill models of student learning – high level overview.

representation of posterior distribution. A more pragmatic approach to skill estimation is Performance Factor Analysis [5], which computes the skill estimate as a linear combination of the number successes and failures of a student. This approach can be extended to take into account ordering of attempts and time intervals between them [2, 4].

Which type of model is better depends on the learning situation. Binary skill models assume a sudden switch from unlearned to learned state. Such assumption is appropriate mainly for fine-grained skills which require understanding or insight (such as "addition of simple fractions"). Models with continuous skill assume gradual increase of skill. This is appropriate either for modeling coarse-grained skills (e.g., "fractions" as a single skill) or for situations where gradual strengthening happens (e.g., memorizing facts).

3. EXPERIMENTS

To analyze the described models and relations between them we performed experiments with simulated data. We generated simulated data by one of the models and then analyzed the generated data using both models with binary and continuous skills. For generating data we used 10 scenarios with different parameter settings.

With respect to accuracy of predictions the results show that both types of models bring consistent improvement over baselines like moving average and time decay models [6]. The basic comparison of binary and continuous skill models is also not surprising: each approach dominates in scenarios which correspond to its assumptions. Nevertheless, in many cases the differences are small and the predictions are actually highly correlated.

Models are not used only for predictions, but they may be useful in themselves for system developers and researchers. Plausible and explainable model parameters may be used to get insight into behaviour of tutoring systems and also for "discovery with models" (higher level modeling). Results of our analysis show that in the case when there is a mismatch between source of the data and a model, interpretation of parameters may be misleading. As a specific example consider simulated students behaving according to the continuous model with $\theta_0 \sim N(-1, 1), \Delta = 0.2$. Here the fitted BKT guess and slip parameters are 0.24 and 0.16. Intuitive interpretation of BKT parameters would thus suggest high chance of guessing an answer. In the ground truth model, however, chance of guessing converges to zero for unskilled students.

One of the main applications of student models is to guide the behaviour of adaptive educational systems. A typical example is the use of student models for mastery learning – students have to practice certain skill until they reach mastery, the attainment of mastery is decided by a student model. Mastery is declared when a skill estimate is higher than a given threshold. How does the choice of student model and a threshold impact student practice? Our results show that the BKT model is relatively insensitive to the choice of the threshold and that the model provides weak decisions for scenarios with continuous learning, specifically when the learning rate is low. Continuous skill models can provide good decision for all scenarios if used with a good threshold. However, optimal thresholds differ significantly for scenarios with binary skill and continuous skill.

To summarize, our study with simulated data suggests that the choice between models with binary and continuous skill does not seem a key concern as long as we are interested only in predictions of students' answers, but it can have significant impact on parameter interpretation and mastery learning.

4. REFERENCES

[1] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[2] A. Galyardt and I. Goldin. Recent-performance factors analysis. In *Educational Data Mining*, 2014.

[3] Tanja Käser, Kenneth R Koedinger, and Markus Gross. Different parameters - same prediction: An analysis of learning curves. In *Educational Data Mining*, pages 52–59, 2014.

[4] J. Papoušek, R. Pelánek, and V. Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining*, pages 6–13, 2014.

[5] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Artificial Intelligence in Education*, volume 200, pages 531–538. IOS Press, 2009.

[6] R. Pelánek. Time decay functions and Elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.

An Analysis of Response Times in Adaptive Practice of Geography Facts

Jan Papoušek
Masaryk University Brno
jan.papousek@mail.muni.cz

Radek Pelánek
Masaryk University Brno
xpelanek@mail.muni.cz

Jiří Řihák
Masaryk University Brno
thran@mail.muni.cz

Vít Stanislav
Masaryk University Brno
slaweet@mail.muni.cz

ABSTRACT

Online educational systems can easily measure both answers and response times. Student modeling, however, typically focuses only on correctness of answers. In this work we analyze response times from a widely used system for adaptive practice of geography facts. Our results show that response times have simple relationship with the probability of answering correctly the next question about the same item. We also analyze the overall speed of students and its relation to several aspects of students' behaviour within the system.

1. INTRODUCTION

When students use computerized educational systems, we can easily store and analyze not just their answers and their correctness, but also the associated response times. Response times carry potentially useful information about both cognitive and affective states of students.

Response times have been studied thoroughly in item response theory in the context of computerized adaptive testing, for an overview of used models see [5]. But testing and learning settings differ in many aspects, including response times – for example we would expect students to think for longer time in the case of high stake testing than in practice session (there are differences even between high-stakes and low-stakes testing [2]).

Response times have been used previously in the context of student modeling for intelligent tutoring systems, e.g., for modeling student knowledge in the extension of Bayesian Knowledge Tracing [6] or for modeling student disengagement [1]. But overall the use of response times has been so far rather marginal. In this work we analyze response times from an adaptive system for practice of facts, which is a specific application domain where response times have not been analyzed before.

2. THE USED SYSTEM AND DATA

For the analysis we use data from an online adaptive system `slepemapy.cz` for practice of geography facts (e.g., names and location of countries, cities, mountains). The system uses student modeling techniques to estimate student knowledge and adaptively selects questions of suitable difficulty [4]. The system uses open questions ("Where is Rwanda?") and multiple-choice questions ("What is the name of the highlighted country?") with 2 to 6 options.

The system uses a target success rate (e.g., 75 %) and adaptively selects questions in such a way that the students' achieved performance is close to this target [3]. The system also collects users' feedback on question difficulty – after 30, 70, 120, and 200 answers the system shows the dialog "What is the difficulty of asked questions?", students choose one of the following options: "Too Easy", "Appropriate", "Too Difficult".

For the reported experiments we used the following dataset: 54 thousand students, 1458 geography facts, over 8 million answers and nearly 40 thousand feedback answers.

3. RESULTS

We provide basic analysis of response times, and their relation to student knowledge and to students' behaviour within the adaptive practice system.

3.1 Basic Characterization of Response Times

Distribution of response times is skewed, in previous work it was usually modeled by a log-normal distribution [5]. Our data are also approximately log-normal, therefor as a measure of central tendency we use median or mean of log times.

Response times clearly depend on the type of question and on specific item. Our results for example show, that response times are higher for cities and rivers than for countries and regions (states are larger than cities on the used interactive map and therefore it is easier to click on them). Response times are also on average higher for countries in Asia than in South America (there is larger number of countries on the map of Asia).

For the below presented analysis we use percentiles of response times over individual items – these are not influenced by skew and provide normalization across different items.

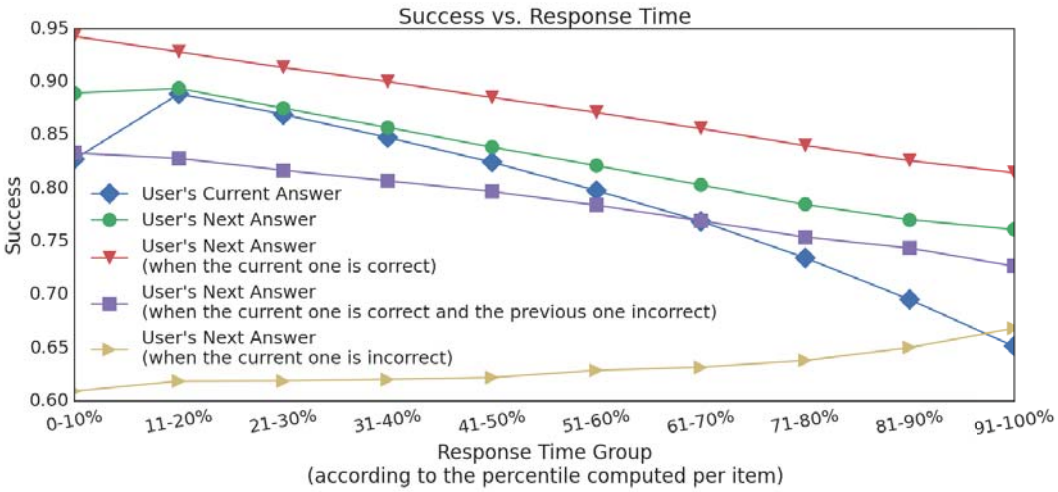


Figure 1: Response times and probability that the (next) answer is correct.

3.2 Response Times and Students Knowledge

Figure 1 shows the relationship between response times and correctness of answers. The relationship between response time and correctness of the *current* answer is non-monotonic – very fast responses combine "solid knowledge" and "pure guessing", long responses mostly indicate "weak knowledge". The highest change of correct answers is for response times between 10th and 20th percentile, i.e., answers that are fast, but not extremely fast.

We get a more straightforward relationship when we analyze correctness of the *next* answer (about the same item) based on both the correctness and response time for the current answer. If the current answer is correct then the probability of correct next answer is linearly dependent on the response time – it goes from 95% for very fast answers to nearly 80% for slow answers. If the current answer is incorrect then the dependence on response time is weaker, but there is still (approximately linear) trend, but in this case in the other direction. When the current answer is incorrect, longer response time actually means higher chance that the next answer will be correct!

A limitation of the current analysis is that we do not take into account types of questions (the number of available choices and the related guess factor) or the adaptive behaviour of the system (the system asks easier questions when knowledge is estimated to be low). However, we do not expect these factor to significantly influence the reported results, which quite clearly show that response times are useful for modeling knowledge and that it is important to analyze response times separately for correct and incorrect answers.

3.3 Speed of Students

As a next step we analyze not just response times for single answers, but over longer interaction with the system. Statistics of response times may indicate affective states or characterize a type of student. For this preliminary analysis we have classified students as fast/slow depending on their median response time and we analyzed correlations with other aspects of their behaviour (in similar way and with analogical results we have also analyzed variance of response time). The reported results do not necessary imply direct relationship as they may be mediated by other factors (like difficulty of presented items).

Slower students answer smaller number of questions in the system. In fact the overall time in the system is nearly the same for students with different speeds, i.e., slower students just solve smaller number of questions during this time. Faster students have higher prior skill and are more likely to return to the system to do more practice. In the feedback on question difficulty slower students report more difficult impression. Possible application of these results is incorporation of students' speed into the algorithm for adaptive selection of questions (e.g., by selecting easier questions for slower students).

4. REFERENCES

[1] J. E. Beck. Using response times to model student disengagement. In *Proc. of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, pages 13–20, 2004.

[2] Y.-H. Lee and H. Chen. A review of recent response-time analyses in educational testing. *Psychological Test and Assessment Modeling*, 53(3):359–379, 2011.

[3] J. Papoušek and R. Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Artificial Intelligence in Education*, 2015.

[4] J. Papoušek, R. Pelánek, and V. Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining*, pages 6–13, 2014.

[5] W. J. Van Der Linden. Conceptual issues in response-time modeling. *Journal of Educational Measurement*, 46(3):247–272, 2009.

[6] Y. Wang and N. Heffernan. Leveraging first response time into the knowledge tracing model. In *Educational Data Mining*, pages 176–179, 2012.

Achievement versus Experience: Predicting Students' Choices during Gameplay

Erica L. Snow[1], Maria Ofelia Z. San Pedro[2], Matthew Jacovina[1], Danielle S. McNamara[1], Ryan S. Baker[2]

[1]Arizona State University, Learning Sciences Institute, 1000 S. Forest Mall, Tempe, AZ 85287

[2]Teachers College Columbia University, 525 W 120th St. New York, NY 10027

Erica.L.Snow@asu.edu, mzs2106@tc.columbia.edu, Matthew.Jacovina@asu.edu, Danielle.McNamara@asu.edu, baker2@exchange.tc.columbia.edu

ABSTRACT

This study investigates how we can effectively predict what type of game a user will choose within the game-based environment iSTART-2. Seventy-seven college students interacted freely with the system for approximately 2 hours. Two models (a baseline and a full model) are compared that include as features the type of games played, previous game achievements (i.e., trophies won, points earned), and actions (i.e., iBucks/points spent, time spent on games, total games played). Using decision tree analyses, the resulting best-performing model indicates that students' choices within game-based environments are not solely driven by their recent achievement. Instead a more holistic view is needed to predict students' choices in complex systems.

Keywords

Game-based environments, Modeling, Decision tree analysis

1. INTRODUCTION

Game-based environments often afford fine-grained examinations of patterns in students' behaviors during gameplay and how they are related to cognitive skills and learning outcomes [1,2]. However, such previous work has not examined the driving force behind *why* a student chooses a specific activity or interaction within a game environment. In the current work, we compare two models. The first model is a parsimonious "1-back" model that assumes that students' choices are directly related to (and predicted by) their most recent game choice within the system and their achievements (in terms of the type of trophy won). Thus, if a student is performing well in one activity, they will continue to play that activity (or one similar to it) – *achievement behavior* [3]. The second, full model assumes that students' choices (of game type in this case) are related more comprehensively to a holistic combination of their previous *experiences* within the environment, including the types of games played, game achievements, and actions. This model follows the assumption that students' choices are influenced by a range of factors that is broader than their most recent choice and achievements. This paper is an exploratory study that attempts to answer: *what impacts students' choices within game-based environments?*

1.1 iSTART-2

Our analysis is conducted within the context of the Interactive Strategy Training for Active Reading and Thinking-2 (iSTART-2) system, designed to provide students with self-explanation strategy instruction to improve reading comprehension [1, 4]. After viewing five instructional videos, each covering a reading strategy, students are transitioned to a practice interface in which they can engage with a suite of educational games. Games involve either *generative* or *identification* practice. Generative practice games require students to type their own self-explanations while reading a text. Identification mini-games require students to read self-explanations that are ostensibly written by other students, and select which of the five strategies was used to generate each self-explanation. Students receive feedback about whether their choice was correct or incorrect.

iSTART-2 offers an ideal environment to explore questions about choice within open learning environments because students are free to choose which practice games to play. During each of the practice games, students earn points for writing high quality self-explanations or selecting the correct strategies. Based on students' score at the end of each game, theycan earn trophies (gold, silver, bronze), *iSTART Points*, and *iBucks*. iSTART Points determine students' current level within the system. iBucks are the system currency and can be spent to customize players' avatars, change background colors, or buy access to the identification games. In the current study, they were provided with an abundance of iBucks to allow them to freely interact with all features.

2. METHODS

2.1 Participants and Procedure

The study included 77 students (18-24 years) from a large University in the Southwest US. We conducted a 3-hour session consisting of a pretest, strategy training (via iSTART-2), extended game-based practice within iSTART-2, and a posttest. For our analyses here, we solely examined data from the time students spent in the game-based practice menu of iSTART. Each student spent approximately 2 hours interacting freely within the game-based interface, with his or her actions logged into the iSTART-2 database.

2.2 Development of Machine-Learned Models of Game Choice

To develop models that predict next game choice from previous achievement in an iSTART-2 game, we distilled features from the interaction logs of the 77 students who interacted with iSTART-2. A total of 1,562 action records were created for these 77 students, where each action record had 13 distilled features. Each record was labeled with the current game choice (at time *n*; 1 = identification game, 0 = generative game), having features corresponding to information about previous gameplay actions (at time *n-1*) in either an identification game or a generative game. In developing the two models to predict students' game choice, we employed student-level cross-validation for a decision tree classifier that uses the J48 implementation [5] that builds a

decision tree from a set of labeled training data. The baseline 1-back model included 2 features: previous type of game played, and type of trophy earned on the previous game. The full model included 11 additional features. The features that involved prior gameplay achievements and actions included: the number of iBucks won/spent, the number of iBuck bonus points won/spent, and the number of iSTART points won/spent the previous time the student played that game type. The remaining five features were aggregates of a student's achievements and actions so far: number of trophies achieved, number of generative games played, number of identification games played, average time played in a generative game, and average time played in an identification game.

3. RESULTS

For the 1-back model that predicts game choice based solely on previous game choice and achievement, students in our data set played a total of 1,562 games in iSTART – 1,144 instances of an identification game played and 418 instances of a generative game. The baseline model performed poorly under student-level cross-validation (see Table 1). This results in an imbalance, with precision of 38.46% and recall of 4.78%. The cross-validated A' is 0.603 (correctly predicted a game choice to be an identification game 60.3% of the time) and cross-validated Cohen's Kappa is 0.208 (model's accuracy was only 2.8% better than chance). This baseline model mainly predicts that students who have just played an identification game will select another identification game, regardless of their trophy achievement. It also predicts that many students who have just played a generative game, but did not receive any trophy, will select an identification game next.

Table 1. Cross-validated confusion matrix of baseline model

	Identification Game (True)	Generative Game (True)
Identification Game (Predicted)	1112	398
Generative Game (Predicted)	32	20

The second model resulted in the best-performing J48 tree with six features: (1) type of trophy from previous game played, (2) number of identification games played so far, (3) number of generative games played so far, (4) iSTART bonus iBucks spent in previous interaction, (5) iSTART points won in previous game, and (6) iSTART iBucks spent in previous interaction.

Table 2. Cross-validated confusion matrix of comprehensive model

	Identification Game (True)	Generative Game (True)
Identification Game (Predicted)	1069	125
Generative Game (Predicted)	75	293

This second model performed significantly better under cross-validation, classifying 1194 game choices as identification games, and 368 game choices as generative games (see Table 2), with a precision of 80.45% and recall of 70.10%. Our cross-validated A' and Cohen's Kappa also increased considerably, to A' = 0.907 and Cohen's Kappa = 0.660. Our second model yields a decision tree size of 61, with 34 decision rules (paths from root to leaf). Some examples of rules within this model include:

1) IF a student has at least played one generative game so far, AND spent more than 50 iSTART iBucks, THEN the next game the student will play is an IDENTIFICATION GAME (Confidence: 99.5%).
2) IF in a previous game the student won more than 610 iSTART points in a previous game, but spent 861 or fewer iSTART iBucks in a previous game, THEN the next game the student will play is an IDENTIFICATION GAME (Confidence: 97.0%).
3) IF a student has not played any generative game so far, AND spent no iSTART iBucks in a previous game, AND has received a BRONZE trophy in the previous game played, THEN the next game the student will play is an GENERATIVE GAME (Confidence: 83.33%).
4) IF a student has not played any generative game so far, AND spent no iSTART iBucks in a previous game, AND has received a SILVER trophy in the previous game played, THEN the next game the student will play is an GENERATIVE GAME (Confidence: 100%).

4. DISCUSSION

Results from this exploratory analysis suggest that students' choices in activities do not rely solely on previous game trophy achievement or previous game choice (first baseline model), but instead students' choices seem to be guided by their overall experience and interactions within the system (second comprehensive model). While this finding is not entirely surprising, it does help researchers shed light upon which features in a game-based environment are impacting students' choices. Indeed, there are many factors that impact students' choices within game-based environments. Thus, within environments where students are afforded a high amount of agency, user models will benefit by incorporating a more complete set of interaction features as a means to represent students' game experience more completely. In the future, we will employ Markov analyses in combination with decision tree analysis in an effort to gain a deeper understanding of what drives students' choices within a game-based environment. Although interactions within agency-driven environments are highly complex, this project demonstrates that they are predictable using machine learning algorithms.

5. ACKNOWLEDGMENTS

This research was supported in part by IES (R305A130124) and NSF (REC0241144; IIS-0735682).

6. REFERENCES

[1] Snow, E. L., Allen, L. K., Jacovina, M. E., & McNamara, D. S. 2015. Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers & Education.* 26, (2015), 378-392.

[2] Sabourin, J., Shores, L. R., Mott, B. W., & Lester, J. C. 2012. Predicting student self-regulation strategies in game-based learning environments. *In ITS 2012* (pp. 141-150). Springer Berlin Heidelberg.

[3] Nicholls, J. G. 1984. Achievement motivation: Conceptions of ability, subjective experience, task choice, and performance. *Psychological review*, 91, (1984) 328-342.

[4] Jackson, G. T., and McNamara, D. S. 2013. Motivation and Performance in a game-based intelligent tutoring system. *Journal of Educational Psychology,* 105, (2013), 1036-1049.

[5] Witten, I. H., & Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

How to Aggregate Multimodal Features for Perceived Task Difficulty Recognition in Intelligent Tutoring Systems

Ruth Janning
Information Systems and Machine Learning Lab
University of Hildesheim
janning@ismll.uni-hildesheim.de

Carlotta Schatten
Information Systems and Machine Learning Lab
University of Hildesheim
schatten@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

Currently, a lot of research in the field of intelligent tutoring systems is concerned with recognising student's emotions and affects. The recognition is done by extracting features from information sources like speech, typing and mouse clicking behaviour or physiological sensors. Multimodal affect recognition approaches use several information sources. Those approaches usually focus on the recognition of emotions or affects but not on how to aggregate the multimodal features in the best way to reach the best recognition performance. In this work we propose an approach which combines methods from feature selection and ensemble learning for improving the performance of perceived task difficulty recognition.

1. INTRODUCTION

Some research has been done in the area of intelligent tutoring systems to identify useful information sources and appropriate features able to describe student's emotions and affects. However, work on multimodal affect recognition in this area focuses more on engineering appropriate features for affect recognition than on the problem of aggregating the features from the different information sources in an good way. The usual approach is to use one classification model fed with one input vector containing the concatenated features (maybe reduced by feature selection) like in [3] or using standard ensemble methods on the features of the sources separately like in [4]. In this paper instead we propose to mixing up the different feature types and combining methods from feature selection and ensemble approaches to reach a classification performance improvement compared to using only either methods from feature selection or ensemble approaches. Feature selection methods can be used to reduce the number of features and find good combinations of features. They take advantage of statistical information like correlations. Ensemble methods like stacking use multiple learning models to obtain a better prediction performance.

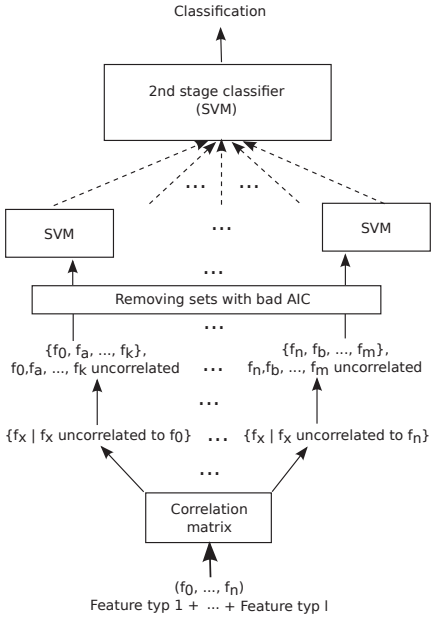


Figure 1: Multimodal feature aggregation approach.

Stacking learns to combine the classification decisions of several single classifiers by a further classifier which gets as input the outputs of the other classifiers.

2. MULTIMODAL FEATURE AGGREGATION

We propose to profit from the advantages of both feature selection methods and ensemble methods. Hence, we combine both (see fig. 1): In a first step the feature vectors of all l feature types are concatenated to reach one input feature vector $(f_0, \ldots, f_n)$. However, there could be dependencies between the different features. Hence, we create the correlation matrix reporting about the correlations between each pair of features. By means of this matrix we extract for each single feature f_y a set $uncorr_y$ containing all other features f_x not correlated to f_y. *Not correlated* means in this case that the correlation value $v_{x,y}$ of the pair (f_x, f_y) in the correlation matrix is near to 0.0, or more explicitly, $|v_{x,y}|$ is smaller than some positive thresh-

Table 1: Classification errors and F-measures.

(1)	SVM applied to amplitude features 31.25% (0.75, 0.59) SVM applied to articulation features 22.92% (0.81, 0.72)
(2)	SVM applied to all concatenated features 27.08% (0.77, 0.67)
(3)	SVM applied to most uncorrelated features 20.83% (0.81, 0.77)
(4)	Stacking applied to $uncorr_y$ sets 20.83% (0.83, 0,74)
(5)	Stacking applied to $uncorr2_y$ sets 16.67% (0.86, 0.80)
(6)	Stacking applied to $uncorr2_y$ sets with best AIC **8.33% (0.92, 0.91)**

old t, i.e. $uncorr_y := \{f_y\} \cup \{f_x \mid t > |v_{x,y}|\}$. The set $uncorr_y$ contains all features uncorrelated to f_y but between the features within this set there could still be correlations. Consequently, we compute for each feature f_y a set $uncorr2_y := \{f_y, f_a, \ldots, f_k\}$ where $f_y, f_a, \ldots, f_k$ all are uncorrelated. These sets $uncorr2_y$ are gained for each feature f_y by sequentially intersecting $uncorr_y$ with the sets belonging to the features within $uncorr_y$, or the intersection respectively. Different to feature selection, our goal is not to create one feature vector with reduced dimensionality but we aim at creating one feature vector per feature which will be fed into an own classifier, to consider each feature and to deliver as many input as needed for the ensemble method. Nevertheless, we remove some of the $uncorr2_y$ sets. The reason is that there is still some statistical information which we did not yet use: the quality of the models using these sets as input. Hence, for each set $uncorr2_y$ we compute the Akaike information criterion (AIC) – indicating the quality of a model. Subsequently, we remove the worse quarter of the sets. The remaining sets are fed into an support vector machine (SVM) each. In the next step we apply a stacking ensemble approach by feeding the outputs, i.e. the classification decisions, of the SVMs into a further SVM, which learns how to generate one common classification decision.

3. EXPERIMENTS

We prove our proposed multimodal feature aggregation approach by experiments with a real data set and multimodal low-level speech features. The data were gained by conducting a study in which the speech of ten 10 to 12 years old German students was recorded and their perceived task-difficulties were labelled by experts. During the study a paper sheet with fraction tasks was shown to the students and they were asked to explain their observations and answers. The acoustic speech recordings were used to gain two kinds of low-level speech features: *amplitude* and *articulation* features. The *amplitude features* ([1]) are taken from the raw speech data, or information about speech pauses respectively: ratio between (a) speech and pauses, (b) number of pause/speech segments and number of all segments, (c) avg. length of pause/speech segments and max. length of pause/speech segments, (d) number of all segments and number of seconds, and percentage of pauses of input speech data. The idea behind this kind of features is that depending on how challenged the student feels, the student makes more or less and shorter or longer speech pauses. The *articulation features* ([2]) are gained from an intermediate step of speech recognition which delivers information about vowels and consonants: ratio between (a) number of silence tags and number of all tags, (b) avg./min. length of vowels/obstruents/fricatives/silence tags and max./avg. length of vowels/obstruents/fricatives/silence tags. The idea behind this kind of features is that depending on how challenged the student is, the student shortens or lengthens vowels and consonants. The data collection resulted in 36 examples labelled with *over-challenged* or *appropriately challenged*, respectively 48 examples after applying oversampling to the smaller set of examples of class *over-challenged* to eliminate unbalance within the data. We conducted a 3-fold cross validation and we applied SVMs with an RBF-kernel and for each SVM used we conducted a grid search on each fold to estimate the optimal values for the hyper parameters. As baseline experiments we applied an SVM separately to both feature types. The classification test errors and F-measures (harmonic mean of *recall* and *precision*) for both classes (*over-challenged*, *appropriately challenged*) are reported in tab. 1, (1). An aggregation of both feature types only makes sense, if we can improve this results. A straight forward way to combine different feature types is to concatenate the features of all types and putting them into one feature vector which serves as input for one classification model. However, this approach does not deliver good results (see tab. 1, (2)) in cases where some features may be correlated and may disturb each other. Hence, one should restrict the input vector by considering the correlations. The results of using only features uncorrelated with most of the other features are shown in tab. 1, (3). As one can see considering correlations helps to improve the classification performance. But still there is space for improvement. Hence, in the following we combine ensemble methods with feature selection which takes into account correlations. In a first step we applied stacking ensemble to the outputs of SVMs applied to the $uncorr_y$ sets (see tab. 1, (4)). However, there could still be correlations within the $uncorr_y$ sets. Hence, as next step we computed for each feature the $uncorr2_y$ set and applied again stacking ensemble, resulting in a classification test error of 16.67 % (tab. 1, (5)). This result is already very good but there is one more statistical information to use: the AIC. We computed for each $uncorr2_y$ set the AIC, threw out the worst quarter of these sets and applied stacking to the remaining sets resulting in a very good classification test error of 8.33 % and F-measures $0.92, 0.91$ (tab. 1, (6)). In summary, the experiments have shown that our multimodal feature aggregation approach is able to improve the classification performance significantly.

4. REFERENCES

[1] R. Janning, C. Schatten, and L. Schmidt-Thieme. Feature analysis for affect recognition supporting task sequencing in adaptive intelligent tutoring systems. In *Proceedings of EC-TEL*, 2014.

[2] R. Janning, C. Schatten, L. Schmidt-Thieme, and G. Backfried. An svm plait for improving affect recognition in intelligent tutoring systems. In *Proceedings of ICTAI*, 2014.

[3] J. Moore, L. Tian, and C. Lai. Word-level emotion recognition using high-level features. In *CICLing*, 2014.

[4] S. Salmeron-Majadas, O. Santos, and J. Boticario. Exploring indicators from keyboard and mouse interactions to predict the user affective state. In *Proceedings of EDM*, 2014.

Teacher and learner behaviour in an online EFL workbook

Krzysztof Jedrzejewski krzysztof.jedrzejewski@pearson.com
Mikolaj Bogucki mikolaj.bogucki@pearson.com
Mikolaj Olszewski mikolaj.olszewski@pearson.com
Jan Zwolinski jan.zwolinski@pearson.com
Kacper Lodzikowski kacper.lodzikowski@pearson.com

All authors work for Pearson IOKI, Dabrowskiego 77, Poznan.

ABSTRACT

In this paper, we present selected findings from our usage analysis of an online English Language Teaching (ELT) workbook. We focus on how teachers assign activities and how learners complete them.

Keywords

ELT, network analysis, time on task

1. BACKGROUND

MyEnglishLab for Speakout Pre-intermediate is an ELT workbook that accompanies a paper textbook. The aim of the product is for the teacher to assign auto-graded homework. On average, about 10 practice activities are assigned by the teachers within a week, with a 30% chance of assigning more than the average. Speakout consists of twelve units that cover 90-120 hours of teaching. Each unit contains about thirty assignable activities centred around grammar, vocabulary, listening, reading and writing. This paper is an exploratory study about how teachers assign such activities and how learners complete them.

2. TEACHER PROGRESSION

2.1 Method

To analyse how teachers progress through units within Speakout, we wanted to show which pairs of units were assigned together. By assigning a unit we mean assigning at least one activity from that unit. In Figure 1 (created using Gephi [1]), a node represents teachers who assigned at least one activity in a given unit. The edges represent those teachers that, having assigned some activities in one unit, moved to another unit. A thicker edge means two units were assigned together more frequently (by more teachers). For example, 185 teachers assigned both Unit 1 and Unit 2. The thickness and length of each edge refers to normalised co-appearance (geometric mean) calculated after Newman [2] as:

$$\frac{n(u_i,u_j)}{\sqrt{n(u_i)\cdot n(u_j)}}$$

where n (…) is the number of teachers that assigned activities in all listed units, and u_i is the i-th unit. Different unit types were highlighted for better readability, namely the regular Units 1-6 (U1-U6) and Units 7-12 (U7-U12) are shown separately from Review and Check 1-4 (R&CH1-R&CH4). The role of the former units is to enable regular day-to-day homework practice, while the role of the latter is to allow the learner to review a larger portion of the material from the three previous units before a test.

2.2 Results

Figure 1 shows that there is no prominent community structure. Teachers tend to focus on smaller chunks of material, especially Units 1-3 and Units 7-9. Figure 2 shows that teachers assign either the regular Units or just the Review and Check units, rarely both. There are more connections between the Review and Check units themselves than between the regular Units. For example, more teachers assign Review & Check 3 together with Review & Check 4 than they assign Units 10-12 together with Review & Check 4.

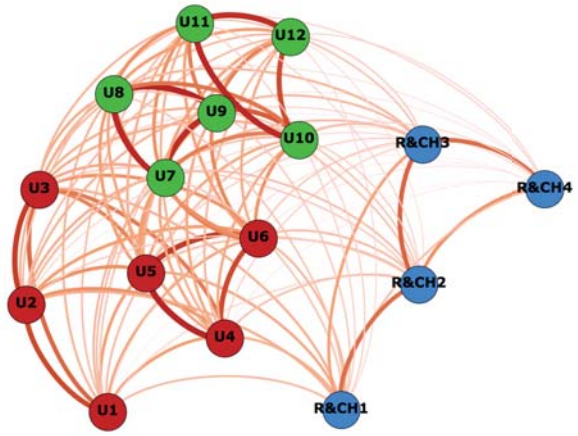


Figure 1. Network graph of relations between units in Speakout Pre-intermediate with edge as a normalised value (geometric mean)

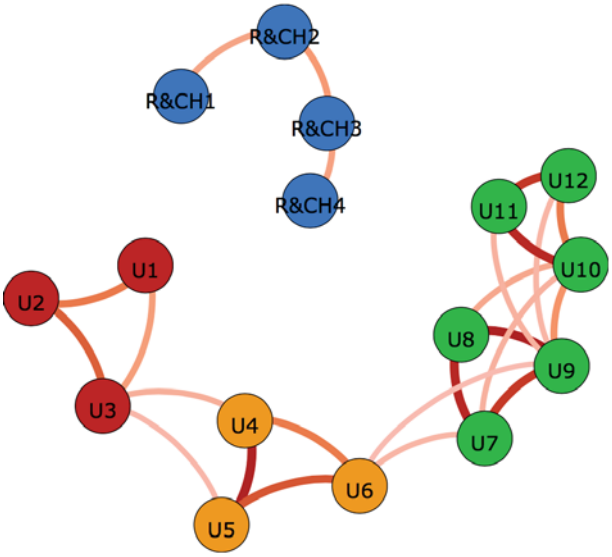


Figure 2. Network graph of relations between units in Speakout Pre-intermediate with edge as a normalised value (geometric mean); only the 24 strongest edges shown

3. QUESTION TYPE AND TIME SPENT

When it comes to learners, we wanted to analyse the time needed for completing a language-learning activity. Speakout contains 15 main question types. Figure 3 (created using RStudio [3]) shows that for most of them the average time spent on the first submission of an activity is of the order of 3 minutes. Learners spend the least time on multiple choice activities (about 1.5 minutes), and most time on jumble words activities (over 4 minutes). We stress that these times do not necessarily correspond to the *optimal* duration it takes a learner to complete all the questions within such an activity, which needs future exploration.

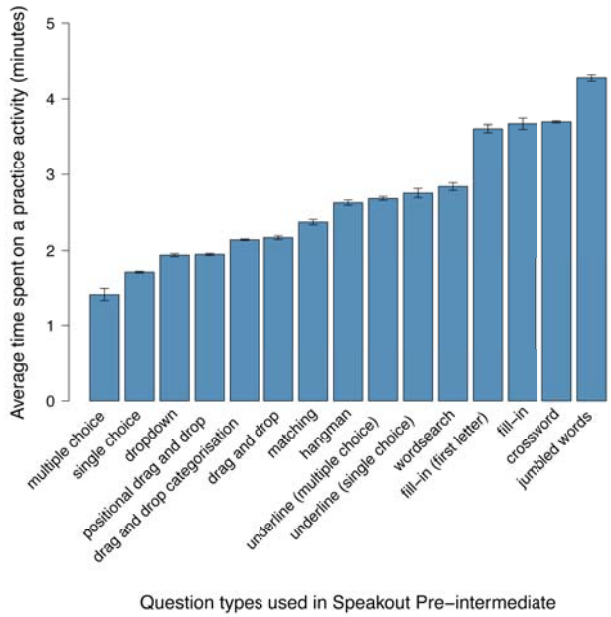


Figure 3. Geometric average of time spent on completing an activity of a given type, with 95% confidence intervals.

Due to space constraints, we present only one figure that presents a question type in more detail, namely *fill-in* (gap completion).

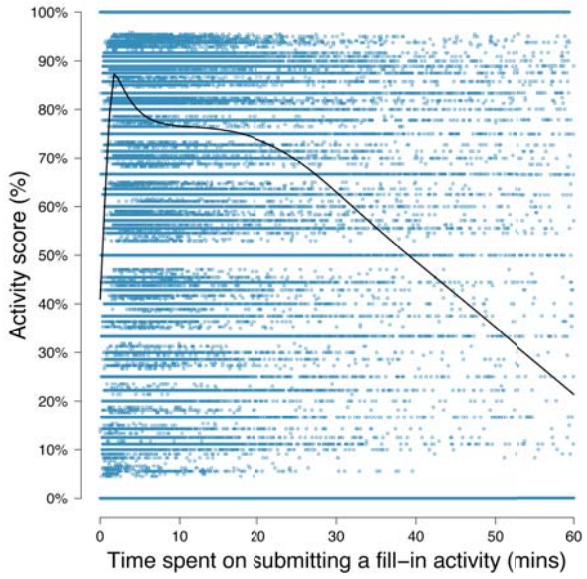


Figure 4. Correlation between the time in which a learner submits a fill-in activity and the score received for that activity; cutoff at 60 minutes

Figure 4 shows that, except for the solid lines at activity score 0% and 100%, most of the observations are placed in the top left part of the plot. The smoothed line shows a peak in activity score at 1.5 minutes spent on a fill-in activity, after which the score visibly decreases. This means many learners need 1.5 minutes to submit a simple fill-in activity (for example, without a text or audio) and receive a relatively high score. An analysis of the top four question types that account for about 76% of Speakout activities (fill-in, drag-and-drop, dropdown and single choice – the last three are not shown here) shows that there is a negative correlation between the time spent on activities and the scores received for those activities. On average, the score decreases by about 8% for each 10 minutes spent on the activities with these question types.

4. FUTURE WORK

Regarding teacher usage, our next step is to segment teachers according to course types and institutions. Regarding learner usage, we will investigate if activities consisting of many questions that are completed within a very short time need to be further analysed to identify whether their format encourages guessing or copying.

5. ACKNOWLEDGMENTS

Our thanks to Rasil Warnakulasooriya for his comments on the early drafts of this work and to the Pearson English MyEnglishLab Team.

6. REFERENCES

[1] Gephi, The Open Graph Viz Platform. Retrieved March 30, 2015, http://gephi.github.io.

[2] Newman, M. 2010. *Networks*. Oxford Scholarship Online. DOI=10.1093/acprof:oso/9780199206650.001.0001.

[3] RStudio. 2012. *RStudio: Integrated development environment for R.* Retrieved March 30, 2015, http://www.rstudio.com

Skill Assessment Using Behavior Data in Virtual World

[1]Ailiya, [2]Chunyan Miao
The Joint NTU-UBC Research Centre
of Excellence in Active Living for the Elderly (LILY)
Nanyang Technological University
Singapore
{[1]ailiya, [2]ascymiao}@ntu.edu.sg

[3]Zhiqi Shen, [4]Zhiwei Zeng
School of Computer Engineering
Nanyang Technological University
Singapore
[3]zqshen@ntu.edu.sg
[4]zzeng001@e.ntu.edu.sg

ABSTRACT
Highly interactive game-like virtual environment has gained increasing spotlight in academic and educational researches. Besides being an efficient and engaging educational tool, virtual environment also collects a lot of behavior data which can be used with Educational Data Mining (EDM) techniques to assess students' learning competencies. In this paper, we propose an assessment system that seamlessly integrates EDM techniques with functionality and affordance of a virtual environment to assess students' learning competency through analyzing their behavioral data and patterns. The virtual environment can record not only students' learning outcome, but also their detailed learning process information, which has the potential to depict the full set of students' learning activity. We also propose a set of metrics which can be used for judging students' Self-Directed Learning skills and how these metrics can be evaluated computationally by capturing students' behavioral data in a virtual environment. The field study, which is conducted in Xinmin Secondary School in Singapore, preliminarily illustrates the effectiveness of our approach.

Keywords
Educational Data Mining; Virtual Environment; Competency Assessment; Self-Directed Learning

1. INTRODUCTION
In the fast changing and increasingly globalized society, students nowadays need to become more conscious, controlled, independent and active in their learning. The new requirements of education urge the creation of new assessment approaches. Besides being an efficient and engaging educational tool, virtual environment also collects a lot of behavior data which can be used with Educational Data Mining (EDM) techniques to assess students' learning competencies. Many researchers have worked in this area [1-3]. The system that we proposed is based on a full-scale 3D virtual environment to assess students' learning competencies. Among all kinds of learning competencies, we focus on Self-Directed Learning (SDL) competency in our research study because it is among the most important learning competencies students need to excel in the knowledge society of the 21st century [4]. SDL skills are important indicators of students' learning competencies as they are the fundamental philosophy behind life-long learning. The proposed system uses Evidence Centered Design (ECD) approach to assess students' SDL competency through analyzing their behavioral data in virtual learning environment. With the Competency Model, Evidence Model, and corresponding Task Model, the system can provide opportunities for students to elicit behavioral indicators of certain SDL skills. These behavioral indicators can be used for assessing the skill levels which cannot be discerned from traditional academic assessment. Moreover, we conducted a pilot study in Xinmin Secondary School Singapore to demonstrate how to evaluate the SDL metrics. The study illustrates the preliminary effectiveness of our approach.

2. MODELING SDL SKILLS
The overall system architecture consists of two main modules: the Virtual Singapura II (VS-II) System and Assessment Automation module. VS-II System is a full-scale 3D virtual world to promote intelligent agent mediated learning. As an open environment, VS-II allows students to explore and learn in a self-directed manner. By recording student's behaviors in the virtual environment, the system provides a convenient and effective setting to elicit students' behavior evidence of their learning skills through the whole learning process.

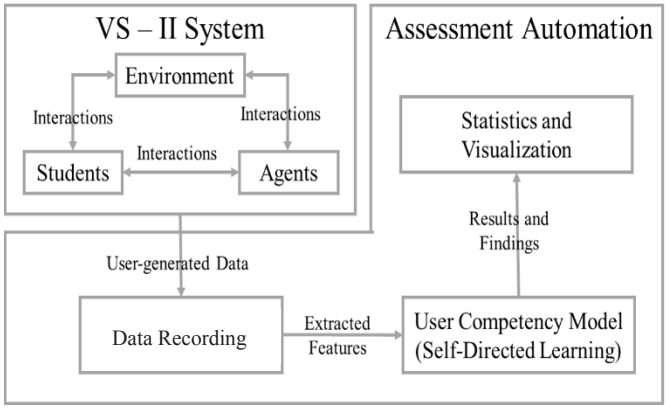


Figure 1. System Architecture for Assessing Students' Learning Competency.

The Assessment Automation module has three sub-modules as shown in Figure 1. The first module – Data Recording meticulously records a wide range of student learning behavior data. There are totally 78 types of events being tracked in the system, and the data collected in the virtual environment consists of three categories: 1) Student learning behavior data, such as locations, timestamps, mouse clicks, etc. 2) Student learning achievement data, such as collected items, fulfilled missions, etc. 3) Student knowledge data, such as correctness of responses, hints required, etc. The second module – User Competency analyzes students' behavioral data through Evidence Centered Design (ECD) approach. Evidence Centered Design (ECD) is the framework for assessment that makes explicit the interrelations among substantive arguments, assessment designs, and operational processes [5]. Similar to the approach Shute has adopted in her study [6], we utilize ECD methodology in our system design to track and interpret students' behavioral data to assess students' SDL competency. The system is designed in a

three-layered model. The three layers are: 1) ***Competency Model*** identifies what should be assessed in terms of skills. The competence of Self-Directed Learning (SDL) is denoted as C_1, where C_1 consists of three aspects of skills $\mathbf{S}^{C_1}$, and $\mathbf{S}^{C_1} = \{S_1, S_2, S_3\}$, where S_1 denotes Ownership of Learning, S_2 denotes Management and Monitoring of Own Learning, and S_3 denotes Extension of Learning. 2) ***Evidence Model*** identifies behaviors that demonstrate the skills defined in 1). The essential student behavioral indicators for SDL are defined as $\mathbf{B}^{S_i} = \{\text{behavioral indicators of } S_i\}$, where $i \in \{1, 2, 3\}$. 3) ***Task Model*** identifies the tasks that would draw out behaviors defined in 2). Let T = {tasks completed by students in the learning environment}, and $\mathrm{T} = \{T_1, T_2, \ldots, T_L\}$**.** Each task T_i is an n-tuple, which consists of an ordered list of learning activities. Let $T_i = (A_1, A_2, \ldots, A_n)$, and $A_i \in \mathbb{A}$ denotes a learning activity. $\mathbb{A}$ is the set of learning activities and each A_i is atomic and cannot be further decomposed into other learning activities.

In our implementation, we focus on the assessment of SDL skills in one of its three aspects, "Management and Monitoring of Own Learning". We illustrated the assessment process by emphasizing one of the skills of SDL competency,S_2, i.e. Management and monitoring of own learning skills. This skill is defined with three behavioral indicators. For each behavioral indicator, we designed several evidence variables to capture a student's performance (as Figure 2). The Last module Statistics and Visualization module visualizes all the results and findings through our user interface.

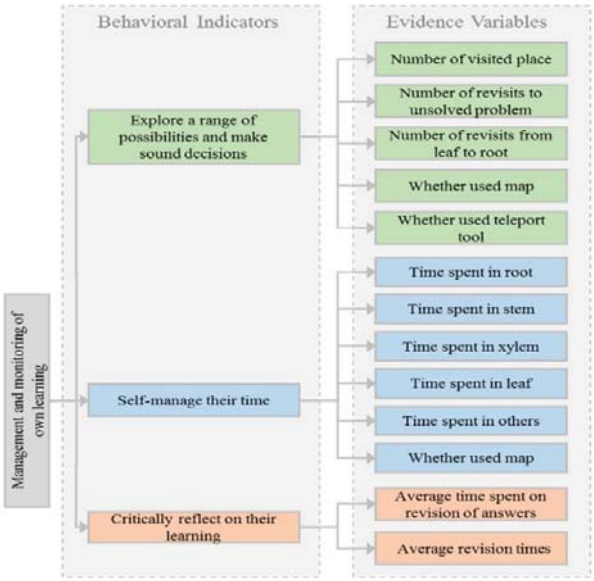


Figure 2. List of Behavioral Indicators and Evidence Variables on Management and Monitoring of Learning.

3. PILOT STUDY AND FINDINGS

The assessment prototype has been deployed in Xinmin secondary school in Singapore. The field study on one side aims to examine whether the whole system is technically workable (data transmission, real time data recording, network setting, client response, etc.), on the other side aims to examine whether the SDL skills can be identified among students with students' real behavioral data. 26 students from Secondary 2 (age 12-13) have participated in our study. In order to get the benchmark, we collected students' SDL skills markings from three of their teachers, and calculated the average scores on each perspective. We also let students fill in a SDL self-report questionnaire.

Significant results have been found. For time management, students who used the virtual map to plan their learning path completed a significantly higher number of learning tasks during the same sessions as compared to those who did not. About 50% of the students in the group with the virtual map completed 4 learning tasks, while for the other group, most students (close to 60% of them) only managed to complete 2 or 3. The average number of learning tasks completed by the group with the map is 3.83, while that of the other group is 2.5. Also, students who more tended to rely on the mobility tools provided in the game (i.e. the teleporting gates, the virtual passport, etc.) tended to limit themselves in terms of self-exploring wide range of possibilities. In contrast, students who were more selective of the tools tend to explore more widely and make better decisions. The correlation coefficient between mobility tool usage and the form teacher's assessment of individual students' exploration skills is -0.5404, indicating a strong negative relationship. These findings support that our system is promising in identifying useful learning behavior metrics, and also has the capability to identify different SDL skills from different behavior patterns.

4. CONCLUSIONS

This paper proposed a virtual environment enabled assessment system for assessing student's SDL skills through personal learning behavior informatics. We provided a set of tools from theoretical models to system implementations to analyze student's behavior data and managed to evaluate the connections between behavioral indicators and student's SDL skills. The proposed three-layered model bridges the gap between definitions of SDL skills and how they can be quantified and evaluated computationally. The seamless integration with VS-II system enables the collection of students' behavioral data in the virtual environment. With the application of educational data mining, the Assessment Automation module analyzes collected behavioral data, consolidates and presents the findings graphically. In the future work, with more and more student data collected, we will gradually refine the benchmarks of student skills and improve the whole assessment process.

Acknowledgement. This research is supported in part by Interactive and Digital Media Programme Office (IDMPO), National Research Foundation (NRF).

5. REFERENCES

[1] Barab, Sasha, et al. "The Quest Atlantis Project: A socially-responsive play space for learning." *The educational design and use of simulation computer games* pp. 159-186. 2007.

[2] DiCerbo, Kristen E. "Detecting Game Player Goals with Log Data." *American Educational Research Association* (2014).

[3] C. Dede, J. Clarke, D. J. Ketelhut, B. Nelson, and C. Bowman, "Students' motivation and learning of science in a multi-user virtual environment." *American Educational Research Association Conference*. 2005.

[4] Sabourin, Jennifer L., et al. "Understanding and predicting student self-regulated learning strategies in game-based learning environments." *International Journal of Artificial Intelligence in Education* 23. no.1-4, pp. 94-114.2013.

[5] R. J. Mislevy, L. S. Steinberg, and R. G. Almond, "Focus article: On the structure of educational assessments," *Measurement: Interdisciplinary research and perspectives,* vol. 1, no. 1, pp. 3-62, 2003.

[6] V. J. Shute, "Stealth assessment in computer-based games to support learning," *Computer games and instruction,* vol. 55, no. 2, pp. 503-524, 2011.

Pacing through MOOCs: course design or teaching effect?

Lorenzo Vigentini
Learning & Teaching Unit
UNSW Australia,
Lev 4 Mathews, Kensington 2065
+61 (2) 9385 6226
l.vigentini@unsw.edu.au

Andrew Clayphan
Learning & Teaching Unit
UNSW Australia,
Lev 4 Mathews, Kensington 2065
+61 (2) 9385 6226
a.clayphan@unsw.edu.au

ABSTRACT

Despite the original tenets about openness and participatory characteristics of MOOCs [1], the majority of MOOCs are delivered in a semi-structured asynchronous way bridging the strong structure of traditional courses -signposted by lectures, tutorials/seminars and activities/assignment deadlines- and open courseware in which student are able to select their own learning paths and goals. Looking at the activity of students in three different MOOCs delivered on the Coursera platform, we considered the effects of different course design to observe variations in the way students pace through the courses. The analysis (in progress) suggests that the course design and the mode of teaching strongly influence the way in which students progress and complete the courses. However, more research needs to be done on the individual variations and on the supporting mechanisms which could be put in place to scaffold students' development of their own learning paths and matching their intended goals.

Keywords

MOOCs, learning design, behavioural analysis, learning

1. INTRODUCTION

Following Gartner's hype cycle [2], MOOCs are currently in the 'sliding into the trough' phase, quickly moving into a consolidation stage, which should lead to the establishment of best practices. This is evident also in the research domain, in which MOOCs have taken centre stage in the recent LAK and Learning@scale conferences. Despite the hype of big data in education and the potential associated with the ability to collect and analyse large amount of information about students' learning behaviours, one of the biggest limitation in the field are the lack of systematicity in the creation of MOOCs –perhaps with the exemption of the limitations of the various platforms- and the lack of strong collaborations leading to sharing data across the sector. As mentioned in [3], at most, researcher might have access to a few MOOCs to analyse; this is echoed in the recent call for a special issue of the JLA (Siemens) to open up and describe large datasets in order to enable research. Yet, the biggest limitation in many published works is a full description of the context, i.e. the course design and philosophy behind it -which is the first stage of any data mining process in the industry-standard CRISM-DM model [4].

Even though the philosophies behind the MOOCs movement range from the instructivist (xMOOC, [5]) to the social-constructivist (cMOOC, [6, 7]), a key assumption is that most MOOCs are built as a 'course': normally there is an instructor/facilitator, a set of resources, activities, support and other participants; content can be curated by instructors or shared among participants. As Cormier [8] put it, a MOOC is 'an event' which provides an opportunity for participants 'to connect and collaborate' and to 'engage with the learning process in a structured way'. But, if it is an *event* and it is *structured*, then the way in which *it* is designed is fundamental and the design is what trumps the teacher role and/or presence. From an academic development's perspective, not only the way in which elements and components are selected and structured makes a difference, but also the philosophy of teaching behind how the course should be delivered drive the learners' experiences.

2. DIFFERENT COURSE DESIGN

At our university, a large, public, research-intensive university in Australia, one of the key reasons to enter the MOOC space was to be able to experiment with pedagogical innovation, learn from it and bring it back to mainstream (i.e. what we do on campus). The selection of courses to be delivered is driven by the awareness of a different target audience, disciplines and the ways in which academics imagined the best ways of teaching a course at scale. Here we only refer to the first 3 courses completed: INTSE (Introduction to System Engineering), LTTO (Learning to Teach Online) and P2P (From Particles to Planets -physics) which are broadly characterised in the table below.

Table 1. Overview of courses

	INTSE	LTTO	P2P
Target group	Engineers	Teachers at all levels	High school and teachers
Course length	9 weeks	8 weeks	8 weeks
Total videos	110	224	98
Total quizzes	10	22	42
Assignments	7	3	2
Forums	54 (14 top level)	105 (17 top-level)	63 (15 top-level)
Design mode	All-at-once	All-at-once	Sequential
Delivery mode	All-at-once	Staggered	Staggered
Use of forums	Tangential	Core activity	Support
N in forum	422	1685	293
Tot posts	1361	6361	1399
Tot comments	285	2728	901
Registrants	32705	28558	22466
Active students[1]	60%	63%	47%
Completing[2]	4.2% (0.3% D)	4.4% (2.4 D)	0.7% (0.2%)

1. Active students are those appearing in the log; 2. Completing are those who achieve the pass grade or earn Distinction (D)

At the surface all three course lean toward an instructivist approach in which the content is essential. However, the educational developers supporting the design ensured that each course was characterised by a mix of content, activities, support tools and evaluation. There are some key differences by design: the way in which content is released; the way in which the course is taught; the function of activities and forums. In INTSE and LTTO all content is released at the start all together, however in LTTO the teaching occurred in a staggered way with regular announcements and feedback videos in response to the top voted comments in each week. P2P used a sequential release of content every week with a staggered delivery and interaction. The activities focus on self-test in INTSE and P2P, while in LTTO these had a teaching function structuring personal development and reflection in the forums. Finally forums were not the focus of the course in INTSE, but had an important role in LTTO and as support in P2P.

3. RESULTS

3.1 Patterns of activity

As it can be seen from the charts some patterns are quite evident. For the P2P course (figure 1), which was designed and delivered on a week-on-week basis, the darker diagonal shows that students are following the course in a linear fashion. LTTO (figure 2) shows a tendency to follow activity along the diagonal. However, this pattern is reduced by individuals who jump between sections/components in the same week (earlier in the course rather than later). In the INTSE (figure 3), patterns are a lot more diluted: in the use of content (videos) the stronger patters occur in the first week, last week and in part across the diagonal. The forums don't seem to have a time-based dependency and the quizzes follow the diagonal and are more frequent in the last week of the course, it is evident that the majority of students tend to follow a fairly linear pattern. Further analysis will be required to test the significance of these patterns, but this early visualization clearly suggest that there is an interaction between the design and delivery of the course and that despite the freedom of determining their learning paths, students like the pacing provided by instructors.

3.2 'Ontrackness' and dedication

In their analysis [3] 'on-trackness' is defined as 'the degree to which students cohere with the recommended syllabus'. Similar metrics have been used in learning analytics as signals for possible support/interventions in order to reduce dropout (i.e. attendance, timely submission etc.). In sequential courses this is simple to identify, however when all the material is available at once, this could be less meaningful. Figure 4 shows the patterns in the three courses by mapping the weeks in which a resource is expected to be used (i.e. design) and when it was actually used. Once again the linear pattern around the diagonal for P2P clearly show how participants follow the course week-on-week; in INTSE and LTTO the videos use are more scattered with quite a few participants looking ahead in the course, but this is not reflected in the quizzes/acitivities and the forums As well as the overview of ontrackness, we have started to consider other metrics, which will require further modelling and analysis. *Dedication* is defined as the regularity of engagement. Given a time period T and the distribution of activity during T, dedication d is the ratio of activity and course length. *Assiduity* is a measure of the patterns of activity over time and it is characterised by the skewness and kurtosis of the distribution of activity. Looking into individual distributions of activity and the relations with other measures will provide a better insight on the individual preferences and how these are related to the teaching and course design.

4. CONCLUSION & DIRECTIONS

Bearing in mind the differences in the cohorts of students taking the courses taken into consideration, which leads to a limited ability to draw conclusions, the striking similarities between the patterns of engagement in the different MOOCs suggests that the method of teaching/delivery is a key element in the way students take a MOOC. The structure of the MOOC 'event' has got a strong impact in the way students engage, but more analysis is necessary to determine the level of flexibility afforded.

At the group level it is apparent that student follow the pace of the course as set by the instructors, however many questions remain open about the effectiveness when it comes to achievement levels. In particular, the goals/intents of students might not be to complete the course and therefore the skipping behaviours could be aligned with what they want to achieve and hard to relate to the measure of success of a MOOC. In fact [9] argue that we need to review and reconceptualise what we mean with student success in this space. More analysis, especially at the individual student level will be necessary to extract meaningful insights.

5. REFERENCES

[1] Dave Cormier and George Siemens. 2010. The Open Course: Through the Open Door--Open Courses as Research, Learning, and Engagement. *EDUCAUSE Review* 45, 4 (January 2010), 30.

[2] Alexander Linden and Jackie Fenn. 2003. Understanding Gartner's hype cycles. *Strategic Analysis Report N^{o} R-20-1971. Gartner, Inc* (2003).

[3] Tommy Mullaney and Justin Reich. 2015. Staggered Versus All-At-Once Content Release in Massive Open Online Courses: Evaluating a Natural Experiment. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*. L@S '15. New York, NY, USA: ACM, 185–194. DOI: http://dx.doi.org/10.1145/2724660.2724663

[4] Colin Shearer. 2000. The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing* 5, 4 (2000), 13–22.

[5] C. Osvaldo Rodriguez. 2012. MOOCs and the AI-Stanford Like Courses: Two Successful and Distinct Course Formats for Massive Open Online Courses. *European Journal of Open, Distance and E-Learning* (January 2012).

[6] George Siemens. 2005. Connectivism: A learning theory for the digital age. *International journal of instructional technology and distance learning* 2, 1 (2005), 3–10.

[7] Stephen Downes. 2008. Places to go: Connectivism & connective knowledge, Innovate.

[8] Dave Cormier. 2009. *What is a MOOC?* YouTube (2009). https://www.youtube.com/watch?v=eW3gMGqcZQc, accessed April 201

[9] Jennifer DeBoer, Andrew D. Ho, Glenda S. Stump, and Lori Breslow. 2014. Changing "Course" Reconceptualizing Educational Variables for Massive Open Online Courses. *EDUCATIONAL RESEARCHER* 43, 2 (March 2014), 74–84. DOI:http://dx.doi.org/10.3102/0013189X145230

Integrating a Web-based ITS with DM tools for Providing Learning Path Optimization and Visual Analytics

Igor Jugo Božidar Kovačić Vanja Slavuj
Department of Informatics
University of Rijeka,
Radmile Matejčić 2, Rijeka, Croatia
+38551584711
ijugo@inf.uniri.hr bkovacic@inf.uniri.hr vslavuj@inf.uniri.hr

ABSTRACT
We present an improved version of our web-based intelligent tutoring system integrated with data mining tools. The purpose of the integration is twofold; a) to power the systems adaptivity based on SPM, and b) to enable teachers (non-experts in data mining) to use data mining techniques on a daily basis and get useful visualizations that provide insights into the learning process/progress of their students.

Keywords
Web based intelligent tutoring system, data visualizations, visual analytics.

1. INTRODUCTION
Our proposed solution to objectives put forth in [5] is the integration of our web-based ITS with standalone data mining tools Weka[3] and SPMF[2]. We developed an integration module that enables continuous communication with the DM tools without implementing any specific algorithm into our application or changing the original DM code. The architecture of the integrated system is displayed in Figure 1. Functionalities that rely on data mining results for students and teachers are marked with asterisks. We will elaborate on these in the next sections. Our web-based intelligent tutoring system (ITS) provides a platform for learning on ill-defined domains [4] i.e. domains that consist of a number of knowledge units (KUs) that do not have a set order in which they have to be learned, but instead the system relies on a domain expert to define the structure of the domain. The learning process is started by selecting a KU to which the system responds by displaying the various types of learning materials created by the teacher. Afterwards, the student proceeds to the assessment module. The system will first ask the student a question about the KU that was learned, followed by an initial question for every KU that is below the current KU in the domain structure created by the teacher. In this way the system checks whether the student understands all the underlying concepts. This list of KUs is currently the same for all students. We aim to make this part dynamic (see Section 3) in order to make the system more adaptive and increase the efficiency of the whole system. If the student offers an incorrect answer to any of the initial questions, he/she is transferred to learning that KU and the whole process is repeated.

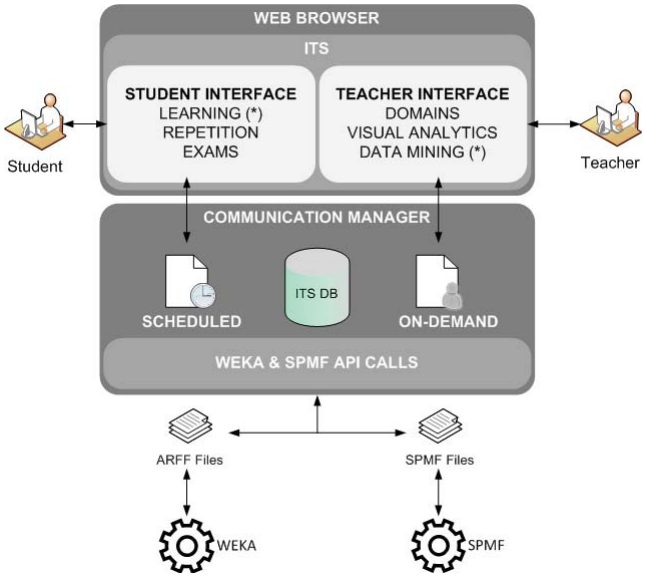


Figure 1. Overall system architecture

No matter how many levels down the hierarchy the student is taken by answering initial questions incorrectly, the system will always return to the starting KU and finish when all the initial questions have been answered. Once the student reaches the KU threshold, the system will stop displaying that KU later in the learning process in order to avoid tediousness and repetition.

2. VISUAL ANALYTICS FOR TEACHERS
At the time of writing the visual analytics section for teachers had a number of visualizations and a clustering section that provide useful insights into the activity of the students and the learning process as a whole. When they start the analytics module, teachers are presented with a compact report containing columns on the number of learning and repetition activities the student performed, number of correct, incorrect and unanswered questions, and the total time spent learning. Each of the columns can be expanded into a sortable, searchable, heat mapped table to get a detailed view about the student's activity. Figure 2 represents the expanded report on the number of learning sessions and repetitions for all the KUs in the domain. Another part of the visual analytics module is the chart section. There are a number of

activity charts that can reveal the activity levels of the whole group or individual students (Figure 3).

Figure 2. Detailed report on learning (all KUs,all students)

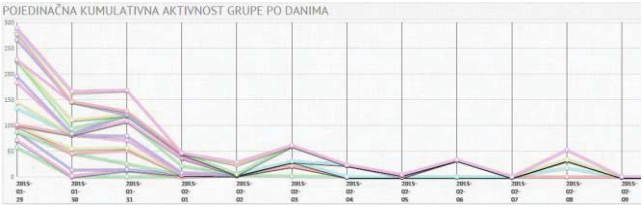


Figure 3. Cumulative group activity by days

The clustering analysis is currently based on a fixed number of features (the ones mentioned in the compact report), but in the next development iteration it will be completely interactive so that the teacher will be able to select features as well as the number of clusters before starting the analysis. When the teacher starts the clustering, the system invokes the communication manager which converts the data to the appropriate file format for either Weka or SMPF, writes the file to the file system and then performs the appropriate API call in the shell command line.

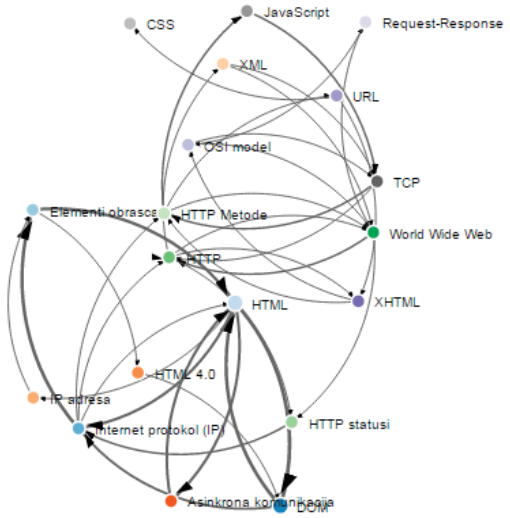


Figure 4. Visualization of student learning paths

The DM tool runs the required algorithm on the data using the sent parameters, and produces the output file. The file is then read, formatted and then returned to the teacher interface where they are displayed as a table with five columns containing cluster names, clusters centroids and students belonging to each cluster. The teachers using the system had no problem identifying inactive students, best students, the average students (largest cluster) and students that were "gaming" the system - students with low number of questions answered and very small amount of time spent learning – they started using the system at the last minute and probably obtained the answers to some questions. This can be confirmed by analyzing the heat maps and activity charts of those students.

3. DM-POWERED PATH OPTIMISATION

The next goal of our research is to create a more adaptive tutoring system in order to: a) increase the quality of learning, b) reduce time needed to acquire the domain knowledge. The set hypothesis is that each student creates a unique path through the structure of KUs. By scheduling a daily analysis of all these paths using SPM algorithms, we can find frequent learning paths. Next, we need to evaluate these paths in order to differentiate between paths that are frequent because a number of students are struggling with a difficult KU without making much progress through the domain from paths that show efficient behaviors that result in significant progress. We are currently developing an algorithm that will perform these evaluations by taking into account a number of learning performance indicators in order to produce a path score. When we get a list of evaluated frequent sequences and students clustered by their activity and effectiveness levels, we can alter the list and order of KUs to be learned in order to help the student follow an optimized path through the knowledge domain. Clustering of students gives us a finer level of granularity so we can offer different modifications to different groups of students. At this moment we run the SPM algorithms to get the frequent patterns and visualize them (Figure 4) using D3JS [1].

4. CONCLUSION

The main advantage of the system is that we can use any of the many SPM and clustering algorithms provided by integrated DM tools. In the future we will complete the SPM based adaptive path optimization component and perform experiments to verify its efficiency.

5. ACKNOWLEDGMENTS

This research is a part of the Project "Enhancing the efficiency of an e-learning system based on data mining", code: 13.13.1.2.02., funded by the University of Rijeka, Croatia.

6. REFERENCES

[1] Bostock, S. M., 2014. D3JS Data Driven Documents. http://d3js.org.

[2] Fournier-Viger, P., et al., 2013. SPMF: Open-Source Data Mining Library. http://www.philippe-fournier-viger.com/spmf/.

[3] Hall, M., et al., 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11, 1.

[4] Lynch, C., et al., 2006. Defining Ill-Defined Domains; A literature survey. In *Proc. Intelligent Tutoring Systems Ill-Defined Domains Workshop*, Taiwan, 1-10.

[5] Romero, C., Ventura, S., 2010., Educational data mining: A review of the state-of-the-art. *Transactions on Systems, Man, and Cybernetics,* , vol. 40, 6, 601-618.

Different patterns of students´ interaction with Mooodle and their relationship with achievement

Rebeca Cerezo
University of Oviedo
Faculty of Psychology
+34 627607021
cerezorebeca@uniovi.es

M. Sanchez-Santillan
University of Oviedo
Computer Science
+34 669 094015
melsanchezsantillan@gmail.com

J.C. Núñez
University of Oviedo
Faculty of Psychology
+34 985 103224
jcarlosn@uniovi.es

M. Puerto Paule
University of Oviedo
Computer Science
+34 689384409
paule@uniovi.es

ABSTRACT
This work tends to broaden the knowledge about the learning process in LMSs from an EDM approach. We examine students' interactions with Moodle and their relationship with achievement. We analyzed the log data gathered from a Moodle 2.0 course corresponding to the different interaction patterns of 140 undergraduate students with the LMS in an authentic learning context. We found out 4 different patterns of learning related to different academic achievement.

Keywords
Learning process, LMSs, Moodle, higher education, log analysis.

1. INTRODUCTION
In traditional learning settings, instructors can easily get an insight into the way that the students work and learn. However, in LMSs, it is more difficult for teachers to see how the students behave and learn in the system [2]. Since learner activities are crucial for effective online teaching-learning process, it is necessary to search for empirical methods to better observe patterns in the online environment. In recent years, researchers have investigated various data mining methods to help instructors to improve e-learning process and systems [1]. As shown in the review of Romero and Ventura [3], a good number of quality works have been conducted with techniques similar to the ones used at this work. Most of them were carried out in laboratory settings with concrete tasks, but just a few in real settings or during an extended period of time [2]. These work aims to go beyond laboratory contexts and researcher-controlled settings. Therefore we set two research questions: 1. Are there sense different patterns of students' interaction when they learn in an LMS in a real context? 2. Are those patterns related to students' final marks?

2. METHODOLOGY
2.1 Participants and procedure
The datasets used in this work have been gathered from a Moodle 2.0 course that enrolled 140 undergraduate university students in a psychology degree program at a state university in Northern Spain. The experience was an assignment in the curriculum of a third year mandatory subject. Students were asked to participate in an eTraining program about self-regulated learning related to the subject's topic. The program was composed of 11 different units that were delivered to the students on a weekly basis. Students get an extra point in their final subject grade if they complete the assignments. We have used 12 actions that make the most sense to represent the students' performance in the particular Moodle course described (See Table 1). The variables selected can be grouped into two different groups: Variables related to effort and time spent working (*Time task*, *Time Span*, *Relevant Actions*, and *Word Forums)* and Variables related to procrastination (*Day's task* and *Day's Hand-in*). *Final marks* were extracted from the performance in the subject that is the grade of the e-Training program and the sum of the grade in an objective final exam of the subject.

2.2 Data Analysis
First, as an exploratory approach to the optimal number of behavioral patterns or clusters in the LMS, the expectation-maximization (EM) algorithm was used. Second, we sought a similar solution to the one provided by EM for the cluster classification but through the k-means algorithm. The objective of these two first steps is to obtain a clustering solution based on coherence among EM and k-means. Through the clustering, we aim to get high similarity intra-cluster and maximize the differences between them. Finally, ANOVA analyses were run to observe if there were differences between the inter-clusters, and the predictive validity of those clusters to predict final marks.

3. RESULTS
After analyzing the data with the EM algorithm, with k-means and with the elbow method, k = 4 was found to be the optimal number of clusters for this sample. Fig. 1 graphically represents the characteristics of the four groups. The second question was to bring up the chances of those patterns being related to students' final marks. For this purpose, an ANOVA analysis was carried out. The results obtained with final marks as the dependent variable and the different clusters the independent ones where $F(3,136) = 13.31$; $p < .00$; η^2_p .227, indicates that there are statistically significant differences between the four student groups in final marks. The post hoc comparisons showed the following statistically significant differences: cluster 1 vs cluster 2 ($d = 0.82$, large effect), cluster 2 vs cluster 4 ($d = 1.43$, very large effect), and cluster 3 vs cluster 4 ($d = 1.01$, large effect).

Table 1. Name of variables considered in the study with their description and extraction method

Name	Description	Extraction Method under *Moodle nomenclature*	Additional information
Variables related to effort and time spent working			
Time Tasks	Total time spent	Sum of the periods between *quiz view/quiz attempt/quiz continue attempt/quiz close attempt* and the next different action	Students have a period of 15 days to complete the tasks.
Time Span	Total time spent working in every unit	Sum of the variables related to the time spent in the three different type of contents: *Time tasks*, *Time Theory* and *Time Forum*	Students have a period of 15 days to work in a declarative knowledge level (*Theoretical contents*), procedural knowledge level (*Practical tasks*), and conditional knowledge level (*Discussion forums*).
Words Forums	Number of words in forum posts	Extracting the number of *forum add discussion* OR *forum add reply* words	Students do not have a minimum/maximum number of words.
Relevant Actions	Number of relevant actions in the LE	Total of relevant actions considered	Actions such as log in, log out, profile updating, check calendar, refresh content, etc. are dismissed.
Variables related to procrastination			
Day's Tasks	How long students wait to check the task since it was made available in the LE (in days)	Date of task *view* since the task was made available	Students have a period of 15 days to complete the tasks.
Day's "hand-in"	The time taken to hand in the task since the task was made available at in LE (in days)	Date of *quiz close attempt* since the task was made available	Students have a period of 15 days to hand in the tasks.

Regarding the comparisons between cluster 1 vs cluster 4 and cluster 2 vs cluster 3, the inter-cluster differences' effect size was medium.

4. DISCUSSION

Four different patterns of learning with different final marks were found in this course; it is interesting how students with very different patterns in the LMS end with a very similar achievement. Cluster 1 is characterized by a small amount of time allocated to work in general but particularly in the practical task. The variables regarding procrastination and the participation in the forums are low, nevertheless, the overall number of significant actions in the LMS is high. Considering that their achievement is medium-low these results may indicate that students in this cluster work quickly but not efficiently. The students in the Cluster 2 could be described as strategic due to the small amount of time and low number of actions in the LMS that led them to very good results. The pattern for working variables is very suitable, too, with a high quantity of time invested in the tasks and they do not procrastinate. Cluster 3 is similar to the previous one in terms of achievement but not in the remaining variables. This group's achievement is a bit lower than Cluster 2's, it could be labeled as medium-high. There is nothing remarkable about procrastination variables, in contrast, the participation in the forums is really low. The number of relevant actions is also the lowest for this cluster; however, the time that they spent in the LMS was the highest. These results may indicate that they are not strategically efficient and do not make the most of the time spent, but they are still ultimately profitable in terms of achievement. Finally, Cluster 4 is characterized by the lowest marks. The most defining characteristic is that they are extreme procrastinators with really low levels in the variables related to the time spent working. Moreover, they make a significant number of relevant actions but do not benefit from them at all, which denotes a maladaptive approach to learning.

On one hand, these results may help an instructor better understand students' learning process, identify at-risk students (e.g., Cluster 1 and 4) and intervene. On the other hand, the information provided by Clusters 2 and 3 could guide the future development of recommendation systems; having a similar performance in terms of achievement the underlying interaction with the LMS denote different patterns that could be modeled by a recommendation systems in very different terms.

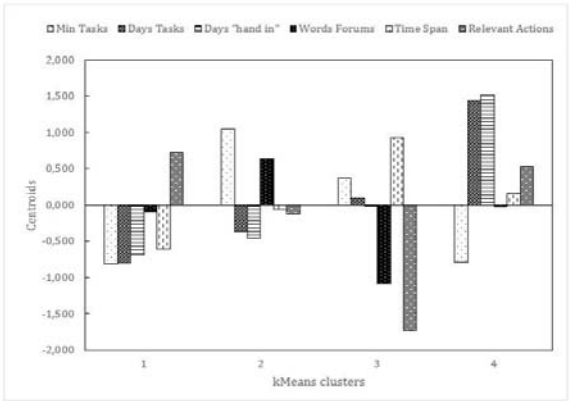


Figure 1. Graphic representation of clustering

5. ACKNOWLEDGMENTS

Our thanks to the Projects TIN2011-25978, EDU2010-16231, GRUPIN14-053 and GRUPIN14-100.

6. REFERENCES

[1] García, E., Romero, C., Ventura, S., & de Castro, C. 2006. Using rules discovery for the continuous improvement of e-learning courses. In *Intelligent Data Engineering and Automated Learning* (Burgos, Spain, September 20 - 23). IDEAL 2006. Springer, Berlin - Heidelberg, 887-895.

[2] Graf, S., & Liu, T. C. 2009. Supporting teachers in identifying students' learning styles in learning management systems: an automatic student modelling approach. *Educational Technology & Society*, 12, 4, 3.

[3] Romero, C. & Ventura, S. 2010. Educational Data Mining: A review of the State of the Art. *IEEE Transactions on Systems, Man, and Cybernetics*, 40, 6, 601-618.

Educational Data Mining in an Open-Ended Remote Laboratory on Electric Circuits. Goals and Preliminary Results

Jordi Cuadros
Lucinio González
IQS Universitat Ramon Llull
Via Augusta 390
08017 Barcelona (Spain)
+34 932 672 000
{jordi.cuadros, lucinio.gonzalez
@iqs.url.edu}

Susana Romero
M. Luz Guenaga
Javier Garcia-Zubia
Pablo Orduña
Universidad de Deusto
Avda. Universidades, 24
48007 Bilbao (Spain)
+34 944 139 000
{sromeroyesa,mlguenaga,zubia,pablo.orduna
@deusto.es}

ABSTRACT
WebLab-Deusto is a learning environment used at the University of Deusto as the landing platform to several remote laboratories currently used in high school and university level courses. One of these remote labs is VISIR, a remote electricity kit that can be used in teaching DC and AC circuits. As happens in any open-ended educational environment, it is difficult to assess the learning effects of this tool. Fortunately the communication between the users and the VISIR remote lab in the Weblab-Deusto leaves behind a set of log information that can be analyzed. This contribution presents our current work-in-progress in analyzing these logs for better understanding the learning processes that take place when using this remote lab.

Keywords
Remote lab, logging, learning, physics, electric circuit

1. INTRODUCTION
WebLab-Deusto [1] is an open-source management system for remote laboratories in development at DeustoTech, Universidad de Deusto since 2001. Its features web and mobile access to several remote laboratories in different topics, e.g. programming or physics.

One of the remote labs that is used through this platform is VISIR [2], a remote laboratory which supports experimentation with electric circuits (see Figure 1).

As is common in using open-ended educational environments, it is difficult for students, teachers and researchers alike to understand and to assess how to use them to improve learning.

Fortunately, the use of VISIR through the WebLab allows collected each of the circuits made by the students and sent to the remote lab for its construction.

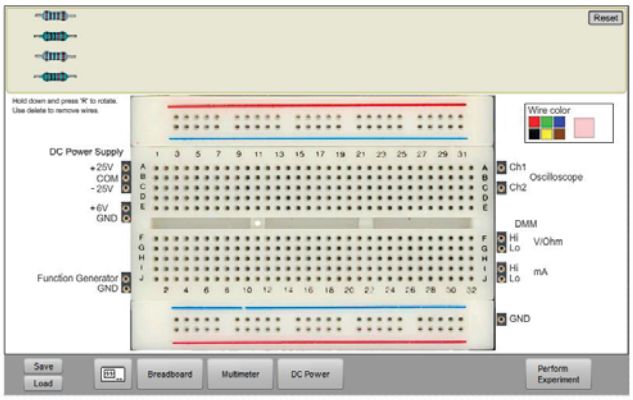


Figure 1. Web interface to VISIR in WebLab-Deusto

This work describes the data collected in WebLab-Deusto for the VISIR remote lab and it presents our efforts to provide (a) a tool for teachers to check students' work, and (b) a toolbox for a quick understanding of the students' activity when the lab is used in medium-to-large class settings.

2. WEBLAB-DEUSTO VISIR DATA COLLECTION
As indicated above, any call to the VISIR remote lab in the WebLab-Deusto system is collected to a database.

Each register in the collected data includes the following fields:

- **studentId**, a key corresponding to each student,
- **sessionId**, a WebLab-Deusto session key,
- **requestTime**, a date/time indicating when the request was made,
- **responseTime**, a date/time indicating when the response was sent back to the web client,

- **queryXML**, the information sent from the client to the remote lab and,
- **answerXML**, the digitized information of the measures collected in the remote lab and sent back to the client.

In this data, the electric circuit made by the user is encoded in character string in the queryXML field. For example, the text "W_X DMM_VHI A11 W_X DMM_VLO A7 R_X A7 A11 10k" indicates that a 10 kΩ resistance is connected to the voltage plugs of the digital multimeter.

3. ASSESSMENT TOOL FOR TEACHERS

The assessment tool for teachers allows selecting a specific call to the remote lab and retrieving in friendly interface the most significant information about the circuit that was constructed and, if it's the case, measured.

This tool, detailed in an earlier publication [3], allows to compare a specific circuit built by a student with a teacher's proposed solution. It automatically evaluates the main characteristics of both circuits and tries to estimate whether both circuits are equivalent.

4. DATA MINING FOR ACTIVITY EVALUATION

The data mining part of the effort implies querying the database for all the actions done by a group of students in solving a predesigned educational hands-on activity.

The results shown here correspond to an educational activity carried in the second semester of the 2013-14 academic year in an introductory physics course in a first-year undergraduate program. It belongs to the teaching of DC circuits, i.e. to the measure of voltage and current in simple DC circuits and Ohm's law. The activity included two 1.5-hour sessions of using the VISIR remote lab.

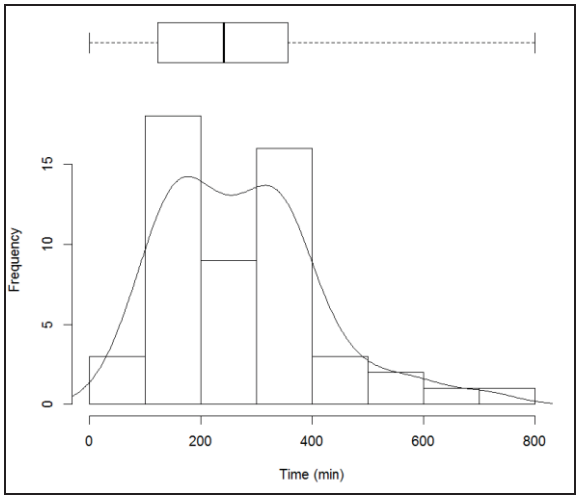


Figure 2. Time spent per student

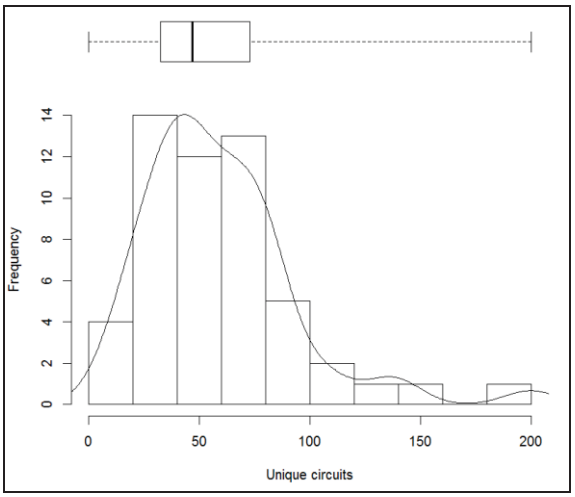


Figure 3. Unique circuits per student

From the pooled data (53 students, 18064 registers, 12114 circuits), a data-based evaluation of the activity has been carried on. For example, the teacher can know the distribution of time-on-task per user (Figure 2), the number of different circuits built per user (Figure 3) or, if required, identify the students who did not take enough profit from the lab session.

Other information that we are currently able to analyze include which circuits are more often built, what measure is attempted in each of them and the correctness of this measure.

5. CONCLUSIONS AND FURTHER WORK

Current logged data in remote laboratories delivers enough information to provide better feedback to students and teachers to support learning in these open environments.

Work is in progress to offer the users of these resources, analytic tools that allow for detecting learning difficulties and affordances for educational improvement.

6. ACKNOWLEDGMENTS

Our thanks to Obra Social "La Caixa" for the funding provided to support this research.

7. REFERENCES

[1] http://weblab.deusto.es/website/

[2] Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., and Lagö, T. 2007. The VISIR project--an open source software initiative for distributed online laboratories. In *Proceedings of the REV 2007 Conference* (Porto, Portugal, June, 2007)

[3] Romero, S., Guenaga, M., García-Zubia, J., and Orduña, P. 2015. Automatic Assessment of Progress Using Remote Laboratories. *International Journal of Online Engineering (iJOE)*, 11, 2, 49-54. DOI= http://dx.doi.org/10.3991/ijoe.v11i2.4379.

Discovering Process in Curriculum Data to Provide Recommendation

Ren Wang
Department of Computing Science
University of Alberta
ren5@ualberta.ca

Osmar R. Zaïane
Department of Computing Science
University of Alberta
zaiane@cs.ualberta.ca

ABSTRACT

Process mining is an emerging technique that can discover the real sequence of various activities from an event log, compare different processes and ultimately find the bottleneck of an existing process and hence improve it. Curriculum data is the history of the courses effectively taken by students. It is essentially process-centric. Applying process mining on curriculum data provides a means to compare cohorts of students, successful and less successful, and presents an opportunity to adjust the requirements for the curriculum by applying enhancement of process mining. This can lead to building recommenders for courses to students based on expected outcome. In this paper we first discover a process model of students taking courses, then, compare the paths that successful and less successful students tend to take and highlight discrepancies between them. The conclusion we reached is that process mining indeed has a great potential to assist teachers and administrators to understand students behavior, to recommend the correct path to students, and at last to enhance the design of a curriculum.

1. INTRODUCTION

The term curriculum often refers to a predefined recommended or mandatory sequence of actions including courses or resources for students. It is designed by a school or a university in order to achieve some educational goals. To maximize this goal, some constraints are frequently imposed, e.g., students must take some specified courses before taking others. Given the liberal approach for selecting courses and taking into account these prerequisites for the courses and the requirements for the programs, students can follow different paths from start to finish. Discovering and understanding the process students follow, or some cohort, such as the most successful learners, can be very indicative to curriculum administrators and can also be the basis for a recommender system to recommend appropriate paths to students in terms of courses to take and in terms of prioritizing the sequence of courses. The common way to analyze educational data is using simple statistics and traditional data mining. However statistics and conventional data mining techniques do not focus on the process as a whole, and do not aim at discovering, analyzing, nor providing a visual representation of the complete educational process [3]. Process mining consists of extracting knowledge from event logs recorded by an information system and is inherent in discovering business process from these event logs, comparing and conforming processes, and providing mechanisms for improvements in these processes[4]. Process mining techniques are often used in the absence of formal description of the process and can provide a visualization with a flowchart as a sequence of activities with interleaving decision points or a sequence of activities with relevance rules based on data in the process.

Some attempts have already been made to exploit the power of process mining in curriculum data, historical data encompassing the sequence of courses taken by students. For instance, the authors of one chapter in [2] give a broad introduction of process mining and indicate that it can be used in educational data. The first paper that proposes to utilize process mining on curriculum data is [3]. The main idea is to model a curriculum as a Colored Petri net using some standard patterns. [1] directly targets curriculum data and brings up a notion called curriculum mining. Similar to the three components of process mining, it clearly defines three main tasks of curriculum mining, which are curriculum model discovery, curriculum model conformance checking and curriculum model extensions.

The application of process mining on curriculum data offers a wide range of possibilities. First it can help the educators understand and make better decisions with regard to the offered curriculum. For example, what is the real academic curriculum? Are there paths seldom used and others more popular? Do current prerequisites make sense? Are the particular curriculum constraints obeyed? How likely is it that a student will finish the studies successfully or will drop out? It can also assist students to choose among different options and even make recommendations to students. For instance, How can I finish my study as soon as possible? Is it more advantageous to take course A before B or B before A? Should I take courses A and B or courses B and D this semester in order to maximize my GPA? Answering such questions to both educators and students can greatly enhance the educational experience and improve the education process. We show in this paper how some of these questions can be answered using the history of courses taken.

2. CURRICULUM DATA

Although the data about courses have already been collected by the Computing Science department of the University of Alberta, we cannot publish any result related to such data due to lack of ethical approval. However, we wrote a curriculum simulator to mimic the behaviors of different kinds of students from the department and be as close as possible to the real data. First, we predefined a set of rules or requirements similar to those in the offered programs in the department. For example, prerequisites, i.e. some specified courses must be taken before the student takes another one. Other requirements include the first and the last course a student must take, mandatory courses, and non-coexisting courses, i.e. if the student takes one course in the group then they cannot take any other course belonging to the same group. Then, we generated students in three categories: the responsible students who always satisfy the course constraints; the typical students who seldom violate course constraint rules; and the careless students who often do not follow the set rules. Moreover, we differentiated the students based on the range of marks they are assigned in courses they take creating clusters of successful and less successful students. We generated the historic courses data for each student adding some probability that a student withdraws from a course giving the course load and previous withdrawing behavior.

3. DATA ANALYSIS

The final goal is to examine what kind of paths successful students tend to take and what is the discrepancy between successful students and less successful students so that we can make recommendations to steer the students to the successful paths. Since we have predefined rules for different types of students in our simulations, the goal is to verify whether we can discern these rules purely from the model we discover by process mining. If we can find the rules from the model, then we are safe to say it is possible to distinguish the "correct path" that can yield the best result by means of process mining, thus a recommendation, that closes the gap among students, can be achieved.

The several process models that were discovered from the curriculum log are close portrayal of real curriculum models in our computing science department. Each model covers the most frequent activity paths, given some thresholds. This is because the model map would be too dense and cluttered to recognize patterns if we present all of them. We added an additional activity at the end of each case to indicate the type of the student. In practice, this type can be any cohort of students such as based on the GPA ranges, based on graduation distinction, withdrawal, or other criteria. To inspect students' behavior patterns in more detail, we further filtered the model with their last activity, i.e., partitioning students based on their type so that we can compare them. The rules we imposed while generating the curriculum data can indeed be easily verified. For the students who seldom violate course constraint rules, the frequent paths appear very similar to those of the first group. However, contrasting the complete graphs of these groups reveals peculiar paths specific to one or the other group. The contrast is even more pronounced when comparing the responsible students and the careless students, as defined in the data. This grouping can be a placement test in some other cases. The categorization can also be done at the end of the paths based on the outcome at the end of the program or the end result for a given course. This allows contrasting the paths taken by successful students with other paths at the end of a program, or comparing the initial paths of students who dropped out of a course to paths leading to the same course taken by those who finish that course. The result of contrasting paths of different cohorts of students stresses out desired and undesired paths specific to some groups, the analysis of which can highlight recommendations for new prerequisites to align new students from a potentially undesired path to the desired one. In the case of drop-outs from courses, this analysis provides insights on the potentially faulty sequence of courses or lack of certain courses in the sequence that lead to higher risk of dropping out. In addition to providing better understanding of the curriculum data and a way to discern between behaviors of different cohorts of students, contrasting between process models from different groups of students presents an opportunity for a course recommender system. By contrasting between the processes followed by students grouped based on their course outcome or based on final GPA, we can find and visualize the sequences of courses that lead to the highest probability of success for a given course. Based on the courses already taken by a student, the system can indicate the options to take that have the highest chance to improve the GPA. Similarly, the system can recommend to take a course before another to maximize outcome. The same data can also be used by administrators to define new prerequisites for courses and thus improve the chance for the adoption of better paths. We are currently building such a recommender system for students. The system would use evidence from historical data to provide comparison of average ranges of prospective marks if a student follows one path or the other when selecting courses.

4. CONCLUSION

Process mining, to discover sequences of courses taken by students, is indeed a powerful tool to analyze curriculum data. By this means, we can visualize and formalize the real paths students actually take, and reveal the underlying patterns such as prerequisites and other constraints. Moreover, conformance in process mining can reveal paths that are unexpectedly not followed by students. Furthermore, contrasting processes from different cohorts of students discloses hidden specificity that we can act upon. Most importantly, contrasting processes provides means to recommend more appropriate sequences of courses to students personalized to their own cases and exposes new insights to administrators.

5. REFERENCES

[1] M. Pechenizkiy, N. Trcka, P. De Bra, and P. Toledo. Currim: Curriculum mining. In *Intl. Conf. on Educational data Mining*, pages 216–217, 2012.

[2] C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker. *Handbook of educational data mining*. CRC Press, 2010.

[3] N. Trcka and M. Pechenizkiy. From local patterns to global models: Towards domain driven educational process mining. In *Intl. Conf. on Intelligent Systems Design and Applications*, pages 1114–1119, 2009.

[4] W. Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

Improving Long-Term Retention Level in an Environment of Personalized Expanding Intervals

Xiaolu Xiong
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
+1-508-831-5000
xxiong@wpi.edu

Joseph Barbosa Beck
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA 01609
+1-508-831-5000
josephbeck@wpi.edu

ABSTRACT
The ability to retain a skill long-term is one of the three indicators of robust learning. Researchers in Intelligent Tutoring Systems (ITS) and Educational Data Mining (EDM) have focused increasing attention on predicting students' long-term retention performance as well as attempting to find effective methods to help improve student knowledge retention. But traditional practices of spacing and expanding retrieval practices have typically fixed their spacing intervals to one or few predefined schedules. In this work, we introduce the Personalized Adaptive Scheduling System (PASS) in ASSISTments' retention and relearning workflow and we have evidence to show that the PASS is helping to improve students' long-term retention performance.

Keywords
Robust learning, spacing effect, knowledge retention, educational data mining

1. INTRODUCTION

1.1 Robust learning and long-term retention
Robust learning is a desirable instructional outcome that goes beyond typical answering a problem correctly immediately following instruction or tutoring. The level of robust learning is assessed by at least one of the three criteria: whether students will be able to transfer their knowledge, whether they will be prepared for future learning, and whether they will retain their knowledge over the long-term [1]. Expanding retrieval practice is often regarded as a superior technique for promoting long-term retention relative to equally spaced retrieval practice [2]. This is specifically crucial to subjects such as mathematics where we are more concerned with students' capability to recall the knowledge they acquired over a long period of time.

1.2 Automatic Reassessment and Relearning System
Inspired by the importance of long-term retention and the design of the enhanced ITS mastery cycle proposed by Wang and Beck [3], we developed and deployed a system called the Automatic Reassessment and Relearning System (ARRS) [4] to make decisions on when to review skills students have mastered in ASSISTments, a non-profit, web-based tutoring system. ARRS is an implementation of expanding retrieval in the ITS environment. ARRS assumes that if a student mastered a skill with three correct responses in a row, such mastery is not necessarily an indication of long-term retention. Therefore, ARRS will present the student with retention tests on the same skill at expanding intervals spread across a schedule of at least three months: the first level of retention tests takes place seven days after the initial mastery, the second level of retention tests 14 days after successfully passing the first retention test, then 28 days, and 56 days. If a student answers incorrectly in one of these retention tests, ASSISTments will give him an opportunity to relearn this skill before redoing the same level of test.

1.3 Personalized Adaptive Scheduling System
Although ARRS helps students review knowledge after a time period, it neither knows a student's knowledge level nor does it have the mechanism to change the retention schedule based on a particular student's performance. Here we formed a hypothesis that we can improve students' long-term retention levels by adaptively assigning students with gradually expanding and spacing intervals over time and we proposed to design and develop such a system, called Personalized Adaptive Scheduling System (PASS), as shown in Figure 1. In the spring of 2014, we enhanced the traditional ARRS with the PASS and deployed it in ASSISTments.

The current workflow of PASS aims to improve students' long-term retention performance by setting up personalized retention test schedules based on their knowledge levels. Here we rely on the *mastery speed* of a skill [4] (number of problems required achieving three consecutive correct responses) as an estimate of the student's knowledge. We retained the ARRS design of 4 expanding intervals of retention tests for each skill; however, PASS alters how tests behave within each interval, especially for the first interval. When a student finishes initially learning a skill, PASS uses his mastery speed to decide when to assign his first level 1 retention test. The longest delay is seven days as students' mastery speed can be as good as three and shortest delay is one day for students who spend seven or more opportunities to achieve initial mastery.

When a student passes the first test, PASS will schedule another test with a longer delay. Once the student passes the seven-day test, he will be promoted to Level 2 with a delay of 14 days. From that point on the intervals are the same as in ARRS system. Note that mastery speed can be extracted from both students' initial learning and relearning processes. Therefore, when a student fails a retention test, a relearning assignment will be assigned to the student immediately and how quickly the student relearns this assignment will be used to set the interval for his next test. The mechanism of Level 2 to Level 4 tests is simpler. When a student fails a retention test, the retention delay will be reduced

to the previous level (e.g., from 56 days to 28 days). It will be increased to the next level if the student passes the delayed retention test.

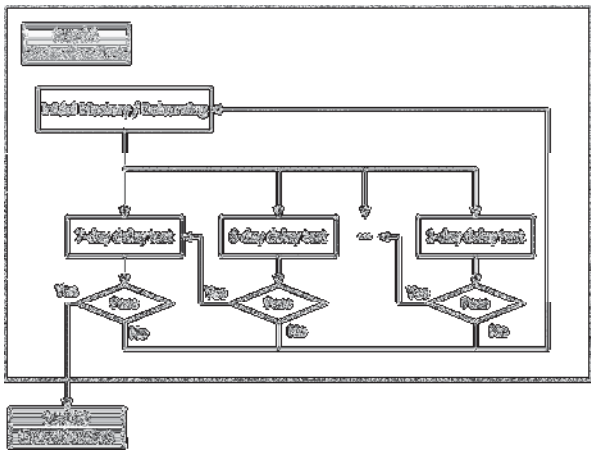


Figure 1. Design of Personalized Adaptive Scheduling System (PASS)

2. IMPACT OF PERSONALIZED EXPANDING RETENTION INTERVALS

A previous study [5] on Level 2 retention tests revealed that students in the PASS condition outperformed those in the ARRS condition and PASS helped to close the performance gap between two groups of students. In fact, in the PASS condition, the long-term performance of medium-knowledge students even slightly outperformed the high-knowledge students.

In this work, we extended our investigation to how students performed on much longer delay after the initial mastery. We collected data that recorded between May 2014 and Feb 2015, which consisted of 4,352 students who have worked on PASS retention tests. We calculated the percentage of correctness on retention tests that within 10 weeks after the completion of a homework assignment, as shown in Figure 2. The data was grouped by the three identified mastery speed bins to represent high-, medium- and low-knowledge students on their initial mastery levels.

It is important to notice that since PASS strictly requires students to achieve a certain level of retention of skills before promoting to the next level of practice, a longer delay doesn't mean a student was working at a higher level of retention test. As we have observed in the previous study [5], some students had to spend four weeks to reach Level 2 retention test while high knowledge level students only need 18 days on average.

The relationship between retention performance and delays in Figure 2 contradicts the general assumption that with strong prior knowledge, performance should decrease as delays get longer. What is seen here is the performance trends got slightly better compared to how students performed at the beginning of PASS workflow. We fitted the performance lines with linear regression trend lines and received positive slopes (0.0057 on average) for all three groups of retention performance. This is can be explained by PASS aggressively assigning short-delay retention tests to weaker students during the first retention level. Another observation is that we again see the persistence of performance differential across three group of students; however, we also noticed the gap between different levels of students was reduced from 12.04% to 7.98% at the end of Week 10. This is further evidence that PASS helps to improve students' retention performance in a classroom context.

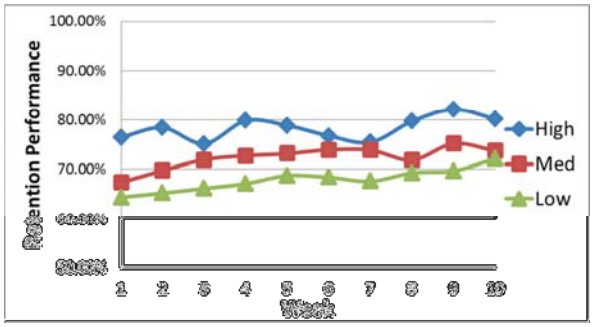


Figure 2. Scatter plot of long-term retention performance in PASS

3. CONCLUSIONS AND FUTURE WORK

This experiment improved the enhanced ITS mastery-cycle model with a personalized expanding interval-scheduling system and explored a simple but effective approach for using ITS to help students achieve better long-term mastery learning. Next, we will work on modeling students' long-term retention performance with data gathered from PASS.

4. ACKNOWLEDGMENTS

We acknowledge funding for ASSISTments from NSF (# 1440753, 1316736, 1252297, 1109483, 1031398, and 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024) grants.

5. REFERENCES

[1] Baker, R. S., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. 2012. Towards automatically detecting whether student learning is shallow. In *Intelligent Tutoring Systems* (pp. 444-453). Springer Berlin Heidelberg.

[2] Hintzman, D. L. 1974. Theoretical implications of the spacing effect.

[3] Wang, Y., & Beck, J. E. 2012. Using Student Modeling to Estimate Student Knowledge Retention. *International Educational Data Mining Society.*

[4] Xiong, X., Li, S., & Beck, J. E. 2013. Will you get it right next week: Predict delayed performance in enhanced ITS mastery cycle. In *The Twenty-Sixth International FLAIRS Conference.*

[5] Xiong, X., Wang, Y., & Beck, J. B. 2015. Improving students' long-term retention performance: a study on personalized retention schedules. In Proceedings of the *Fifth International Conference on Learning Analytics And Knowledge* (pp. 325-329). ACM

Exploring Problem-Solving Behavior in an Optics Game

Michael Eagle, Rebecca Brown, and Tiffany Barnes
North Carolina State University
Department of Computer Science
{mjeagle, rabrown7, tmbarnes}@ncsu.edu

Elizabeth Rowe, Jodi Asbell-Clarke, and Teon Edwards
Educational Gaming Environments (EdGE) @ TERC
{elizabeth_rowe, jodi_asbell-clarke, teon_edwards}@terc.edu

ABSTRACT

Understanding player behavior in complex problem solving tasks is important for both assessing learning and for the design of content. Previous research has modeled student-tutor interactions as a complex network; researchers were able to use these networks to provide visualizations and automatically generated feedback. We collected data from 195 high school students playing an optics puzzle game, Quantum Spectre, and modeled their game play as an interaction network. We found that the networks were useful for visualization of student behavior, identifying areas of student misconceptions, and locating regions of the network where students become stuck.

1. INTRODUCTION

This work presents preliminary results from our attempts to derive insight into the complex behaviors of students solving optics puzzles in an educational games using a complex network representation of student-game interactions. An *Interaction Network* is a complex network representation of all observed student-tutor interactions for a given problem in a game or tutoring system [3]. Professors using *InVis* were successful in performing a series of data searching tasks; they were also able to create hypotheses and test them by exploring the data [5]. *InVis* was also used to explore the behavior of students in a educational game for Cartesian coordinates. Exploration of the interaction networks revealed off task behavior, as well as a series of common student mistakes. The developers used the information gained from the interaction networks to change some of the user interface to reduce these undesirable behaviors [4]. Regions of the network can be discovered by applying network clustering methods, such as those used by Eagle et al. for deriving maps high-level student approaches to problems [2]. This paper reports gameplay data from 195 students in 15 classes collected as part of a national Quantum Spectre implementation study in the 2013-14 academic year.

The Education Gaming Environments (EdGE @ TERC) research group studies how games can be used to improve learning of fundamental high-school science concepts. EdGE games use popular game mechanics embedded in accurate scientific simulation so that through engaging gameplay, players are interacting with digitized versions of the laws of nature and the principles of science. We hypothesize that as players dwell in scientific phenomena, repeatedly grappling with increasingly complex instantiation of the physical laws, they build and solidify their implicit knowledge over time. Previous work for a game *Impulse* used an automated detector of strategies in the game [1]. In this study, we examine how interaction networks can be used to visually measure the implicit science learning of students playing *Quantum Spectre*, a puzzle-style game that simulates an optics bench students might encounter in a high school physics classroom.

2. QUANTUM SPECTRE

Quantum Spectre is a puzzle-style designed for play in browsers and on tablets. Each level requires the player to direct one or more laser beams to targets while (potentially) avoiding obstacles. For each level, an inventory provides the player with access to resources, such as flat and curved (concave, convex, and double-sided) mirrors, (concave and convex) lenses, beam-splitters, and more, that can be placed and oriented within the puzzle and that interact with and direct the laser beams in a scientifically accurate manner. When the appropriate color laser beam(s) have reached all the targets, a level is complete. The player earns three "stars" if the puzzle has been solved in the fewest possible moves, two "stars" for a low number of extra moves, and one "star" for any solution. Each placement or rotation of an object on the game board counts as one move. A player can go onto to the next level as soon as a puzzle is complete, regardless of the number of moves used, but the stars system provides an incentive for level replay and an understanding of the puzzle's solution. The game includes a range of scientifically accurate optical instruments and related science concepts, but for the research, three key scientific concepts were identified: The Law of Reflection; Focal Point and Focal Length of Concave Mirrors; and Slope.

3. RESULTS AND DISCUSSION

To construct an Interaction Network for a problem, we collect the set of all solution attempts for that problem. Each solution attempt is defined by a unique user identifier, as well as an ordered sequence of interactions, where an interaction is defined as {initial state, action, resulting state}, from the start of the problem until the user solves the prob-

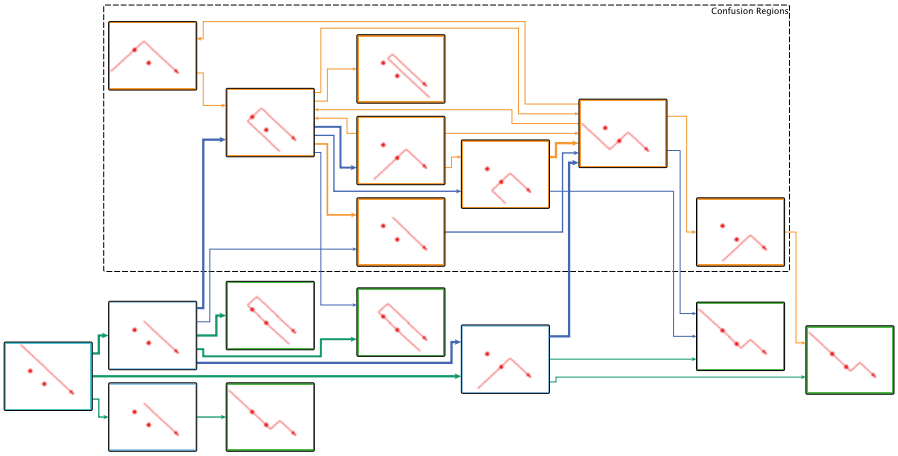


Figure 1: The approach map for problem number 18. This is a high-level view at student approaches to this puzzle. The vertices represent sub-regions of the overall interaction network. Vertices are colored according to their game "star" score, with green being the optimal score, blue the less optimal, and orange for very suboptimal states. The approach map is capturing students with poor approaches to the problem, these regions are indicated by the dotted line.

lem or exits the system. The information contained in a *state* is sufficient to precisely recreate the tutor's interface at each step. Similarly, an *action* is any user interaction which changes the state, and is defined as {action name, pre-conditions, result}. We chose to use only objects the player can interact with. We ignore the distinction between objects of the same type, so the order of placement does not matter. An example state could be {Flat_Mirror(4,1,90), Flat_Mirror(5,5,180)}: which would be a state describing two mirror objects with the first two numbers representing the X and Y coordinates and the last representing the mirrors angle.

The full graph of every state space and every action taken was large, complex, and difficult to interpret in terms of player understanding. In order to provide a high-level view that game designers and instructors could use to gauge players' mastery of game concepts, we clustered states using the Approach Map method from Eagle et al. [2]. The interaction network for problem 18, which had over 1000 unique states, is concisely represented as 17 region-level nodes as seen in figure 1.

This image is a simplified representation of the game board, with a mirror drawn in every location where a mirror was placed by an edge entering the cluster. "Active" pieces (the piece that was moved or rotated to enter the cluster was considered active for that move) were shown in blue, and inactive pieces (any pieces that remained unmoved on the board during that action) were in black. The intention was to show a milestone for each cluster: by looking at how each student who entered a cluster got into that cluster, the reader could trace a given path from cluster to cluster and get an idea of how the students on that path had progressed through the puzzle.

Using the approach map we are able to derive an overview of the student behaviors. Several of the derived regions represent poor approaches to solving the problem, this mirrors the results from Eagle et al. [2]. The region vertices are particularly useful for discovering the locations where students transfer into the confusion regions, as these highlight the places where student approaches contain misunderstandings. These results support the use of approach maps and interaction networks for use in this game environment. In future work we will look for differences in student performance on pre and posttest measures to see if there are differences in overall approach that are predicted by pretest score or can predict posttest score.

4. ACKNOWLEDGMENTS

We are grateful for NSF/EHR/DRK12 grant 1119144 and our research group, EdGE at TERC, which includes Erin Bardar, Jamie Larsen, Barbara MacEachern, and Katie McGrath.

5. REFERENCES

[1] J. Asbell-Clarke, E. Rowe, and E. Sylvan. Working through impulse: assessment of emergent learning in a physics game. *Games+ Learning+ Society*, 9, 2013.

[2] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. *Proceedings of the Seventh International Conference on Educational Data Mining*, 2014.

[3] M. Eagle, D. Hicks, P. III, and T. Barnes. Exploring networks of problem-solving interactions. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.

[4] M. Eagle, M. Johnson, T. Barnes, and A. K. Boyce. Exploring player behavior with visual analytics. In *FDG*, pages 380–383, 2013.

[5] M. W. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks.

Simulating Multi-Subject Momentary Time Sampling

Luc Paquette
Teachers College, Columbia U.
525 W. 120th Street
New York, New York 10027
+1 212 678 3854
paquette@tc.columbia.edu

Jaclyn Ocumpaugh
Teachers College, Columbia U.
525 W. 120th Street
New York, New York 10027
+1 212 678 3854
jo2424@tc.coumbia.edu

Ryan S. Baker
Teachers College, Columbia U.
525 W. 120th Street
New York, New York 10027
+1 212 678 8329
baker2@exchange.tc.columbia.edu

ABSTRACT
This paper presents software for examining measurement error in Momentary Time Sampling—an interval time sampling method commonly used in research domains (e.g., classroom observations) where continuous recording is not feasible. The Parameters for Optimizing Scientific Sampling Using Momentary-time-sampling Simulator (POSSUMS) produces Monte Carlo simulations (based on user-specified values) and automatically generates statistics relevant to understanding the extent to which measurement error may be expected within multi-subject design parameters.

Keywords
Momentary Time Sampling, Monte Carlo simulation, student behaviors, classroom observations.

1. INTRODUCTION
Educational research and other investigations of behavior often rely on sampling procedures when continuous observation is not viable [4]. As researchers in Educational Data Mining (EDM) have sought training labels for affect/engagement detectors to study the effects of student classroom behaviors on long-term outcomes, they have also relied on sampling procedures (e.g., [6]'s review). These include momentary time sampling (MTS), where researchers divide the observation session into intervals, recording whether a particular behavior occurs at the end of each. MTS, also known as *instantaneous time sampling* or *point sampling*, proves more accurate than similar techniques, including whole interval recording (WIR, where behavior is only recorded if it was present throughout the sampling interval) or partial interval recording (PIR, where behavior is recorded as present if it occurs at any time during the interval) [8]. Still, MTS is prone to substantial measurement error for some study designs [5].

Measurement error in MTS—the difference between *actual* and *observed* values for specific behaviors—is influenced by a large number of interacting factors [14], but research focuses on the duration of the behavior being observed and the length of the observation interval (e.g., [1]). Although the method does not introduce bias, the sometimes substantial variation in apparently transient measurement error has led to highly conservative suggestions, including [11], who suggest that MTS should only be used after continuous observations first determine typical values for factors known to influence MTS measurement error.

A different approach to dealing with the uncertainty in MTS is to model measurement error through simulation (see extensive review in [14]), sometimes to study particular conditions and other times to make more general recommendations (e.g., [10], [13], [12], [3]). However, existing simulators [7] have focused on single-subject designs, which is inadequate for modeling measurement error in observation systems where an observer is coding multiple students in the same session (e.g., BROMP [6], a common method for EDM research; but also classroom observation schemes used by many public schools in the U.S.). In this poster, we present a freely available simulator that addresses this gap: the Parameters for Optimizing Scientific Sampling Using Momentary-time-sampling Simulator (POSSUMS).

2. Prior Research
Prior research has shown that measurement error in MTS may be induced by a number of interacting factors, including: (a) the *sampling interval* (how often observations are recorded) (b) the *observation session's length*, (c) *bout-length* (the duration of each event/behavior being observed) and (d) *prevalence*, the percentage of an observation session that a behavior occupies (as [6] and [14] review). Previous research with simulations has led to practical recommendations, such as specific limits on sampling intervals (e.g., less than every 60 seconds [2] or 120 seconds [9]), or more general suggestions (e.g., sampling intervals must be shorter than mean bout-length [1], [14]), but these recommendations are based on simulations involving single-subject design. That is, these are recommendations for estimating the amount of time that a single research subject (e.g., a student) spends engaging in a particular behavior (e.g., on-task conversation) over a given observation session (e.g., an hour long class). They have not been demonstrated to be appropriate for estimating prevalence in multi-subject research designs (e.g., the amount of time that students in a particular classroom spend engaged on on-task conversation over the course of a class session). What's more, simulators that are currently publically available for single-subject design (e.g., [7]) require programming skills, limiting their use to researchers familiar with that programming language.

3. POSSUMS 1.0
In this paper, we present POSSUMS 1.0: a java-based tool that allows researchers using MTS in multi-subject design to run Monte Carlo simulations to study potential measurement error. POSSUMS, which is freely available on the 1st author's webpage (http://www.columbia.edu/~lp2575/tools.html), allows users to set parameters which it models, automatically generating metrics needed to understand potential sources of error. In the sections that follow, we present the user interface and the output.

3.1 User Interface
POSSUMS presents users with an interface that allows them to set several relevant parameters. As shown in Figure 1, users first add

target behaviors to be observed, specifying projected bout-length and prevalence for each. They then specify how many subjects (students) will be coded and the length of the observation window. (These values are used to run a Monte Carlo simulation that represents the *actual*, continuous values that might be found *en vivo*.) The user also specifies multiple sampling intervals, in seconds, which are used to simulate MTS *estimates*—the values that would be obtained based on sampling at those intervals across multiple subjects. Finally, the user indicates how many simulations should be run.

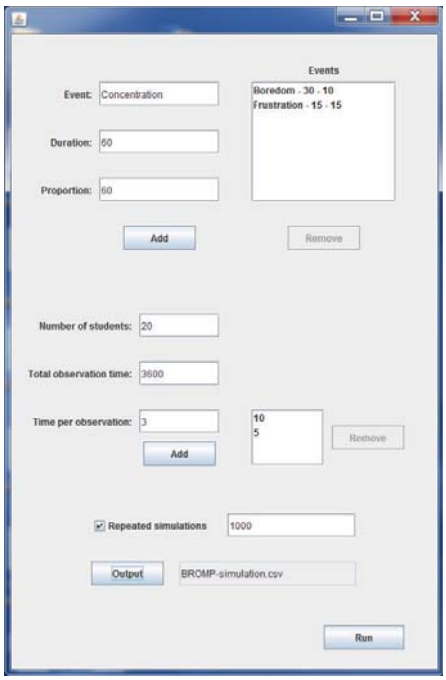


Figure 1. POSSUMS 1.0 User Interface

3.2 Output

POSSUMS 1.0 outputs to .csv files, which import easily into Excel or other widely-used analyses tools. The exact format depends on how many simulations are run. When only a single simulation is executed, the output file summarizes how many time each target behavior was observed, providing percentages for each behavior's contribution to the total observations at the classroom and the student level. These files also contain a detailed list of the behaviors associated to each student at each second in the simulated observation period. When multiple simulations are executed, the output file is different, providing summary measures that average across all simulations. Those values include the average and standard deviations for the number of time the behavior was observed across simulations, and the average and standard deviation for the percentage of observations for each behavior across simulations.

4. Discussion/Conclusions

Educational research, like other domains that sometimes require observational research, has long relied on sampling methods to estimate actual values. As EDM research begins to make use of observational methods to estimate the prevalence of relevant behaviors or events in classroom settings (cf. [6]), it is important that researchers understand possible sources of measurement error. Because this error in MTS appears to be influenced by a large number of factors working in concert, to date efforts to quantify it have focused on single-subject design (e.g., [7], [14]). Unfortunately, these studies are insufficient for understanding measurement error in many observational studies of classroom conditions, which involve multi-subject designs. POSSUMS 1.0 represents a step forward in this effort, simulating pertinent field conditions and automatically generating metrics needed to understand potential sources of error.

5. ACKNOWLEDGMENTS

Thanks to James Pustejovsky, Elizabeth Tipton, Didith Rodrigo, and Sweet San Pedro, for help understanding issues presented here. Special thanks to Stefan Slater, who motivated the name.

6. REFERENCES

[1] Ary, D., & Suen, H. K. 1983. The use of momentary time sampling to assess both frequency and duration of behavior. *Journal of Behavioral Assessment*, **5**, 143–150.

[2] Brittle, A. R., & Repp, A. C. 1984. An investigation of the accuracy of momentary time sampling procedures with time series data. *British Journal of Psychology*, **75**, 481–488.

[3] Fiske, K., Delmolino, L. 2012. Use of discontinuous methods of data collection in behavioral intervention: Guidelines for practitioners. *Behavior Analysis in Practice*, **5**, 77–81.

[4] Mudford, O. C., Taylor, S. A., & Martin, N. T. 2009. Continuous recording and interobserver agreement algorithms reported in the *J. of Applied Behavior Analysis* (1995–2005). *J. Applied Behavior Analysis*, **42**, 165–169

[5] Murphy, G., & Goodall, E. (1980). Measurement error in direct observations: A comparison of common recording methods. *Behaviour Research & Therapy*, **18**, 147–150.

[6] Ocumpaugh, J., Baker, R.., Rodrigo, M. 2015. *Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) 2.0 Tech. & Training Manual.* NY, NY: Teachers College, Columbia U. Manila, PH: Ateneo Laboratory for the Learning Sciences.

[7] Pustejovsky, J. E., & Runyon, C. 2014. Alternating Renewal Process Models for Behavioral Observation: Sim.Methods, Software, & Validity Illustrations. *Behav. Disorders*, *39*(4).

[8] Rapp, J. T., Colby-Dirksen, A. M., Michalski, D. N., Carroll, R., & Lindenberg, A. 2008. Detecting changes in simulated events using partial-interval recording & momentary time sampling. *Behavioral Interventions*, **23**, 237–269.

[9] Rhine, R. & Ender, P. 1983. Comparability of methods used in sampling primate behavior. *Am. J. Primatology*, **5**, 1–15.

[10] Rojahn, J., & Kanoy, R. 1985. Toward an empirically based parameter selection for time-sampling observation systems. *J. Psychopathology & Behavioral Assessment*, **7**, 99–120.

[11] Sanson-Fisher, R., Poole, A, & Dunn, J. 1980. An empirical method for determining an appropriate interval length for recording behavior. *J. App. Behavior Analysis*, **13**, 493–500.

[12] Suen, H., & Ary, D. 1986. A post hoc correction procedure for systematic errors in time-sampling duration estimates. *J. Psychopathology & Behavioral Assessment*, **8**, 31–38.

[13] Wilson, R., Jansen, B., & Krausman, P. 2008. Planning & assessment of activity budget studies employing instantaneous sampling. *Ethology*, **114**, 999–1005.

[14] Wirth, O., Slaven, J., & Taylor, M. 2014. Interval sampling methods and measurement error: A computer simulation. *Journal of Applied Behavior Analysis*, *47*(1), 83-10.

Analyzing Students' Interaction Based on their Responses to Determine Learning Outcomes

Fazel Keshtkar
Southeast Missouri State University
One University Plaza, Cape Girardeau, MO, USA
fkeshtkar@semo.edu

Jordan Cowart
Southeast Missouri State University
One University Plaza
Cape Girardeau, MO, USA
jrcowart1s@semo.edu

Ben Kingen
Southeast Missouri State University
One University Plaza
Cape Girardeau, MO, USA
bwkingen@semo.edu

Andrew Crutcher
Southeast Missouri State University
One University Plaza,Cape Girardeau, MO, USA
alcrutcher1s@semo.edu

ABSTRACT
Online learning platforms such as Moodle and MOOC (Coursera, edX, etc) have become popular in higher education. These platforms provide information that are potentially useful in developing new student learning models. One source of information provided by these platforms is in the form of student interaction with one another, instructors, and the platform itself. These interactions contain various activities such as: participation in forum discussion, how frequently a student is logged into their account, and frequency of reading posted activities, etc. Using Data Mining techniques, namely clustering algorithms to find students with similar behavior patterns, our goal is to develop a student model that can be conducted by learning these interaction patterns. In doing so, we aim to develop a method by which to provide students with different guidelines and instructions that will help to improve their performance. This research is in progress and our data include Moodle online courses in computer science in different semesters.

Keywords
Online Learning, Student Behaviors, Student Outcomes, Moodle, Data Mining, Clustering, Educational Data Mining

1. INTRODUCTION
Detecting students' performance is one of the most crucial tasks in online learning and educational data mining (EDM), a task which falls under the scope of classification/clustering or other algorithms. Various learning methods have been applied to detect course results and academic performance with each learning algorithm performing differently with different datasets [4]. The No Free Lunch Theorem states that it is difficult to choose a specific model or classification algorithm for this difficult task [2]. Therefore, discovering and applying appropriate methods for a specific dataset should yield a significant improvement in the effectiveness of a given learning algorithm. Our approach will apply learning algorithms based on metadata, as they have proven to be sufficient to address this problem [2]. These meta-learning algorithms have been studied by exploring metadata to adopt suitable algorithm based on data mining and machine learning techniques [5]. In this research, we propose to apply various classifications/clustering models, evaluation measurements, and statistical analysis test to predict the performance of students' learning outcomes based on new dataset. This paper focuses on a portion of our statistical analysis, namely the examination of student response times to professor activity.

2. DATASET
Our dataset contains student and professor metadata from eleven courses over two semesters at Southeast Missouri State University. The metadata is in the form of log data from the online learning platform that the school uses, Moodle. In order to determine which of the features the metadata provides, we have performed rudimentary statistical analysis using SPSS. A basic overview of our dataset is provided in Table 1.

Table 1. Course Overview

Course	Number of Students	Number of interactions	Average Interactions
CS1	12	4281	356.75
CS2	**53**	**14006**	**264.26**
CS3	23	3891	169.17
IS1	33	26682	808.55
IS2	31	20049	646.74
IS3	10	7906	790.60
IS4	13	13311	1023.92
IS5	19	10986	578.21
IS6	30	31433	1047.77
IS7	7	13150	1878.57
UI1	27	7127	263.96

2.1 Data Processing
For this portion of analysis, we analyze CS2 (bold in Table 1.) for students' interaction response times; this was due to the large sample size it provided with respect to the other courses in our dataset. There were five students that failed to complete this course, so they were dropped from the dataset for this particular portion of analysis to prevent data skewing in the later weeks of the class.

3. METHODOLOGY

We propose that applying data mining techniques and statistical analysis of metadata from an online learning platform will allow us to derive insights into student interaction patterns. Using these insights, we theorize that a student learning model can be developed by learning these interaction patterns. In doing so, we aim to develop a method by which to provide students with different guidelines and instructions that will help to improve their performance.

3.1 Feature Selection

Our dataset explicitly provides the following features: the course in which an activity occurred, the time of occurrence, the IP address from which an activity originated, the user which performed the action, the action occured (course, user, assignment, and grade view), and information about the action completed.

There are also metadata that are not explicitly provided in the dataset but can be extracted. For example, our dataset does provide with specifics of the activity that the student is performing (e.g. posting to a forum, content of their posts, etc.) . However, we aware that a student is automatically logged out from their Moodle account if they have not performed an activity within 15 minutes. Using this knowledge, we can then determine when a student is logged out, approximately number of times they login, and the time interval between logins. We are aware that that there may be more metadata hidden within our dataset that maybe found upon closer examination that we plan to consider for future research.

Finally, we consider statistical features that have been extracted. For this portion of the analysis we considered how quickly the students responded to activities made by the professor; these activities include: updates to materials, posting of assignments, and updating student grades. We have computed the average student response time per activity, a sample standard deviation for the response time per activity, the total average response time and a sample standard deviation for the course during the first two weeks and the entirety of the course. We have also computed the top ten activities that resulted in the quickest average response times and the top ten activities that resulted in the slowest average response times.

4. RESULTS AND DISCUSSION

One of our goals was to explore trends in how students interact with their course over the duration of a semester and, more specifically, how quickly they react to activities performed by their professor. We noticed that when a professor interacts with Moodle, they typically perform a lot more than one action. For our statistics, we counted the time it took for each student in the course to respond to the last update to the course page that a professor made in a continuous block of interactions. For each of these interactions, we then calculated the average response time per student and the overall average response time for that particular activity. The average response times per activity are shown in Figure 1. We can see that student activity has fluctuates throughout the semester, but further analysis is needed to determine possible causes for these fluctuations. The only readily explainable peak is activity thirty, which occurred during a five day break.

5. RELATED WORKS

Wang [6] has indicated a need for the examination of log analysis within online learning platforms, namely the examination of

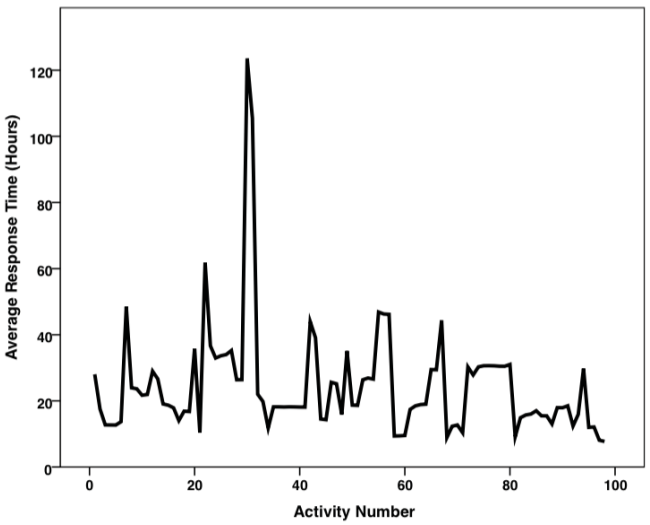


Figure 1. Average response times per activity.

indicators of participation such as use of discussion forums, quiz completion rate, and video usage. The research of Yudelson et al [7]. indicates that finding and analyzing certain sub-populations within a student body can produce a better predictive model than that of examining the entire population; importantly, these sub-populations tend produce a more substantial data footprint [7]. The research of Coffrin et al. indicates that student interactivity and success during the first two weeks of a course strongly related to their outcomes at the end of the course. They also suggested that identifying students based on their patterns of engagement presents the opportunity of tailored feedback to these sub-populations [1].

6. ACKNOWLEDGMENTS

This research is funded by GRFC grant, Southeast Missouri State University.

7. REFERENCES

1. Coffrin, C., Corrin, L., Barba, P., Kennedy, G. Visualizing patterns of student engagement and performance in MOOCs. Proceedings of the Fourth International Conference on Learning Analytics and Knowledge. 2014.
2. Hamälainen, W., Vinni M. Classifiers for educational data mining; Handbook of Educational Data Mining. Chapman & Hall/CRC. 2011.
3. Ho T.K., Basu M. Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):289-300, 2002.
4. Romero, C. and Ventura, S. Data Mining in Education. Wire Interdisciplinary Reviews: Da-ta Mining and Knowledge Discovery, 3:12-27. 2013.
5. Song, Q, Wang, G, Wang, C. Automatic recommendation of classification algorithms based on dataset characteristics. Pattern recognition. 45, 2672–2689, 2012.
6. Wang, Y. MOOC Learner Motivation and Learning Pattern Discovery - A Research Prospectus Paper. Proceedings of the 7th International Conference on Educational Data Mining. 2014.
7. Yudelson, M., Fancsali, S., Ritter, S., Berman, S., Nixon, T., Joshi, A. Better Data Beat Big Data. Proceedings of the 7th International Conference on Educational Data Mining. 2014.

Exploring the Impact of Spacing in Mathematics Learning through Data Mining

Richard Tibbles
Department of Cognitive Science
University of California, San Diego
9500 Gilman Drive
San Diego, California, USA
rtibbles@ucsd.edu

ABSTRACT

Laboratory studies suggest that long term retention of Mathematics learning is enhanced by spaced, as opposed to massed, practice. However, little evidence has been evinced to demonstrate that such spaced learning has a positive impact in real world learning environments, at least partly because of entrenched pedagogy and practice, whereby students are encouraged to engage with Mathematics in a very sequential manner - thus leading to massed learning episodes. Indeed, much educational practice and the structure of Mathematics textbooks lend themselves to massed rather than spaced learning. However, in online learning such spaced practice is possible and more practically achieved. Predicting learner outcomes from data in a popular online Mathematics learning site shows that in this data set spacing seems to have a negative effect on retention at a later time.

Keywords

Mathematics, spaced learning, learning science, online learning

1. INTRODUCTION

Learning efficiently is one of the main drivers of personalized instruction. By ensuring that students engage with material only for as long as they need to in order to master it, intelligent instruction can push students further in less time, allowing outcomes to be improved more rapidly, and also to reduce the risk of boredom and loss of motivation. In addition, retention over longer time scales is important to the goals of Education as a whole. While the old adage "Education is what is left once what is learned has been forgotten" is oft quoted, in many Educational contexts, and in particular Mathematics, the necessity of prerequisite knowledge for learning higher order material means that such forgetting is far less desirable.

Until relatively recently in pedagogical practice (as shown by the design of Mathematics textbooks), it was thought that the most efficient way for a student to learn Mathematics in a way that facilitated later retrieval was *overlearning* - the continued practice of a procedure after mastery has been achieved. This *massed* (as opposed to *spaced*) practice model explains the design of Mathematics textbooks, where, by chapter, exercises are massed by a small number of procedures that need to be applied. By contrast, a spaced learning methodology would require intermingled exercises requiring application of different kinds of procedure, but with procedures recurring multiple times over several study sessions.

Spacing has been a core component of recent advances in our understanding of the Science of Learning. Rohrer and Pashler[7], drawing on work by Rohrer and Taylor[8], identify the empirical support for using such spaced learning episodes in the learning of Mathematics. Rickard et al.[6] examined the role of spacing in promoting retrieval over calculation in mathematics, and spacing of learning has been assessed in the college Mathematics classroom by Butler and colleagues[1]. Both found spacing to have positive effects on Mathematics learning. However, most recent work has focused more on the effect of spacing on declarative fact learning, with much of the successful practical application focused on foreign language vocabulary learning[4][5][9]. If these techniques can be extended to Mathematics learning, then considerable learning gains could be achieved.

Such hypotheses are best tested through a more controlled manipulation of the spacing regime - in the online learning context, using an A/B test common in most website implementations. Exposing some subset of users to a spaced learning regime, while recommending massed learning to the remainder. However, it is also possible to examine the impact of spaced learning in a somewhat more confounded way by looking at spaced learning that has occurred naturally during the course of student engagement.

2. DATA

The data being analysed are logs from Khan Academy's interactive Mathematics exercise platform. Students answer exercises, and are given instant feedback. The data recorded for each attempt includes the exercise type, the instance of the exercise, the answers given by the student, the time the student spent on the page while answering, the time it was attempted, and whether the student used a hint or not.

2.1 Spaced Learning

Khan Academy has attempted to implement spaced learning within its site design mostly derived from the spaced repetition algorithm popularized by Leitner[3]. In the Leitner System cards that have been correctly memorized are pushed back into a later set, whereas incorrectly answered cards are placed into the first set. The first set is reviewed on every cycle, with each set beyond being reviewed one less time per cycle (for N boxes, a cycle will consist of N review sessions).

The variable implementation of this spacing design over time in the Khan Academy site (including the use of A/B testing for various implementations of this spaced repetition algorithm), in addition to the voluntary engagement with the software by student users has served to create a data set with a large variety of spacing schemes (although somewhat confounded by other variables). Using this data, we are conducting a post hoc analysis of spaced versus massed practice. This will help to shed light on the impact of spaced repetition on learning of particular Mathematics skills.

3. ANALYSIS

Recent experimental studies on spaced learning have generally been constructed around one or more temporally separated (by periods of more than a day) study sessions, followed by a further temporally separated recall session, where retention of what has been learned is measured[2]. In order to emulate this design for each student, data, subdivided by exercise, were separated into study sessions (any gaps of a day or more were assumed to constitute a separate study session). In order to have an outcome measure by which to measure student learning, the final session was taken to be the retention session.

3.1 Data Selection

In order to ensure more meaningful comparisons, all student/exercise pairings with only one session associated with them (and therefore no differentiable outcome measure) were discarded, as were students who had made less than ten attempts across all sessions on that particular exercise. A random subsample was chosen for analysis, with data from 13528 students, and a total of 155602 student/exercise pairs. All data were normalized before fitting in order to render model coefficients more meaningful.

4. RESULTS

In order to assess the potential contribution of the effect of spacing, a logistic regression model using L2 regularization (strength parameter set by 10-fold cross validation) was fitted to predict student performance during the retention session. The independent variables included in the model were: mean accuracy across all study sessions, mean accuracy in the most recent study session, total time spent on exercises during study sessions, total number of study sessions, and total number of attempts during study sessions. While the model performed relatively poorly, (achieving approximately 58% accuracy on the test data) similar performance was seen predicting from most subsets of the independent variables. Only total time spent failed to lend any power to the model.

Table 1: Coefficients for Normalized Variables

Mean Study Accuracy	38.25
Recent Accuracy	-15.61
Total Study Time	0.74
Number of Study Sessions	-13.36
Study Attempts	8.59

5. CONCLUSIONS

The results seem to indicate that, at least in the case of the Khan Academy data, that spaced learning does not help with later retention. However, as much of the engagement takes place over relatively short time scales (with the median interval between study and retention being ten days). Further analysis will look at the impact of spaced learning not only on later retention of that skill, but also on learning skills for which the learned skill is a prerequisite. This will allow the impact of spaced learning to be assessed absent the compressed nature of engagement with individual exercises.

6. ACKNOWLEDGMENTS

I acknowledge support from a Small Grant provided by the Temporal Dynamics of Learning Center, an NSF funded Science of Learning Center. I would also like to thank Khan Academy for making the data available for analysis.

7. REFERENCES

[1] A. C. Butler, E. J. Marsh, J. P. Slavinsky, and R. G. Baraniuk. Integrating Cognitive Science and Technology Improves Learning in a STEM Classroom. *Educational Psychology Review*, 26(2):331–340, June 2014.

[2] N. J. Cepeda, E. Vul, D. Rohrer, J. T. Wixted, and H. Pashler. Spacing effects in learning a temporal ridgeline of optimal retention. *Psychological science*, 19(11):1095–1102, 2008.

[3] S. Leitner. *So lernt man lernen*. Herder, 1974.

[4] R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving studentsâĂŹ long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[5] H. Pashler, N. Cepeda, R. Lindsey, E. Vul, and M. C. Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in neural information processing systems*, pages 1321–1329, 2009.

[6] T. C. Rickard, J. S.-H. Lau, and H. Pashler. Spacing and the transition from calculation to retrieval. *Psychonomic Bulletin & Review*, 15(3):656–661, 2008.

[7] D. Rohrer and H. Pashler. Increasing retention without increasing study time. *Current Directions in Psychological Science*, 16(4):183–186, 2007.

[8] D. Rohrer and K. Taylor. The shuffling of mathematics problems improves learning. *Instructional Science*, 35(6):481–498, Nov. 2007.

[9] H. S. Sobel, N. J. Cepeda, and I. V. Kapler. Spacing effects in real-world classroom vocabulary learning. *Applied Cognitive Psychology*, 25(5):763–767, 2011.

Toward Data-Driven Analyses of Electronic Text Books

Ahcène Boubekki
ULB Darmstadt/TU Darmstadt
boubekki@dipf.de

Ulf Kröhne
DIPF Frankfurt/M
kroehne@dipf.de

Frank Goldhammer
DIPF Frankfurt/M
goldhammer@dipf.de

Waltraud Schreiber
KU Eichstätt
schreiber@ku-eichstaett.de

Ulf Brefeld
TU Darmstadt/DIPF
brefeld@cs.tu-darmstadt.de

ABSTRACT

We present data-driven log file analyses of an electronic text book for history, called the mBook, to support teachers in preparing lessons for their students. We represent user sessions as contextualised Markov processes of user sessions and propose a probabilistic clustering using expectation maximisation to detect groups of similar (i) sessions and (ii) users.

1. INTRODUCTION

Electronic text books may offer a multitude of benefits to both teachers and students. By representing learning content in various ways and enabling alternative trajectories of accessing learning objects, electronic text books offer great potentials for individualised teaching and learning. Although technological progress passed by schools for a long time, inexpensive electronic devices and handhelds have found their way into schools and are now deployed to complement traditional (paper-based) learning materials.

Particularly text books may benefit from cheap electronic devices. Electronic versions of text books may revolutionise rigour presentations of learning content by linking maps, animations, movies, and other multimedia content. However, these new degrees of freedom in presenting and combining learning materials may bring about new challenges for teachers and learners. For instance, learners need to regulate and direct their learning process to a greater extent if there are many more options they can choose from. Thus, the ultimate goal is not only an enriched and more flexible presentation of the content but to effectively support teachers in preparing lessons and children in learning. To this end, not only the linkage encourages users to quickly jump through different chapters but intelligent components such as recommender systems [4] may highlight alternative pages of interest to the user. Unfortunately, little is known on the impact of these methods on learning as such and even little is known on how such electronic text books are used by students.

In this article, we present insights on the usage of an electronic text book for history called the mBook [5]. Among others, the book has been successfully deployed in the German-speaking Community of Belgium. We show how data-driven analyses may support history teachers in preparing their lessons and showcase possibilities for recommending resources to children. Our approach is twofold: Firstly, we analyse user sessions to find common behavioural patterns across children and their sessions. Secondly, we aggregate sessions belonging to the same user to identify similar types of users. This step could help to detect deviating learners requiring additional attention and instructional support.

2. THE MBOOK

The *mBook* is guided by a constructivist and instructional-driven design. Predominantly, the procedural model of historical thinking is implemented by a structural competence model that consists of four competence areas that are deduced from processes of historical thinking: (i) the competency of posing and answering historical questions, (ii) the competency of working with historical methodologies, and (iii) the competency of capturing history's potential for human orientation and identity The fourth competency includes to acquire and apply historical terminologies, categories, and scripts and is best summarised as (iv) declarative, conceptual and procedural knowledge.

Imparting knowledge in this understanding is therefore not about swotting historic facts but aims at fostering a reflected and (self-)reflexive way of dealing with our past. The underlying concept of the multimedia history schoolbook implements well-known postulations about self-directed learning process in practice. The use of the mBook allows an open-minded approach to history and fosters contextualised and detached views of our past (cf. [3]). To this end, it is crucial that a purely text-based narration is augmented with multimedia elements such as historic maps, pictures, audio and video tracks, etc. Additionally, the elements of the main narration are transparent to the learners. Learners quickly realise that the narration of the author of the mBook is also constructed, as the author reveals his or her construction principle.

3. METHODOLOGY

For lack of space, we only sketch the technical contribution. We devise a parameterised mixture model with K components to compute the probability of a user session. The

browsing process through chapters is modelled by a first-order Markov chain so that pages are addressed only by their chapter. The category model depends on the chapters as we aim to observe correlations between different types of pages. This may show for example whether galleries of some of the chapters are more often visited (and thus more attractive) than others and thus generate feedback for the teachers (e.g., to draw students attention to some neglected resources) and developers (e.g., to re-think the accessibility or even usefulness of resources). The model for the connection times is inspired by the approach described in [2] to capture repetitive behaviour across weeks. The final model is optimised by an EM-like algorithm.

4. EMPIRICAL RESULTS

In our empirical analysis, we focus on about 330.000 sessions collected in Belgium between March and November 2014 containing approximately 5 million events.

Session-based View: Figure 1 (top) shows the results of a session-based clustering. User sessions are distributed across the clustering according to the expressed behaviour. Clusters can therefore be interpreted as similar user behaviours at similar times. The visualisation shows that all categories are clearly visible for all clusters, indicating a frequent usage of all possible types of resources by the users. Cluster *C6* possesses half of the mass on the weekend of category *text*. This indicates more experienced users who like to form their opinion themselves instead of going to summary pages. The same holds for cluster *C8* that possesses in addition only a vanishing proportion of the *home* category. Small probabilities of category *home* as well as large quantities of category *text* indicate that users continuously read pages and do not rely on the top-level menu for navigation.

User-based View: Our approach can also be used to group similar users. To this end, we change the expectation step of the algorithm so that sessions by the same user are processed together. That is, there is only a single expectation for the sessions being in one of the clusters. Clusters therefore encode similar users rather than similar behaviour as in the previous section.

Figure 1 shows the results. Apparently, the main difference of the clusters is the intensity of usage during working days and weekends. Cluster *C2* for instance clearly focuses on working day users who hardly work on weekends compared to Cluster *C1* whose users place a high emphasise on Saturdays and Sundays. Cluster *C3* contains low frequency users who rarely use the mBook and exhibit the smallest amount of sessions and page views per session. Cluster *C8* contains heavy (at night) users with high proportions of category *text*. In general, we note that transition matrices are consistent between chapters in contrast to the session-based clustering, that is, test takers interact with most of the chapters.

5. DISCUSSION

Our results illustrate potential benefits from clustering learners for instructional purposes. In the first place, the probabilistic clustering approach shows a way how to condense a huge amount of logfile information to meaningful patterns of learner interaction. Classifying a student into one of several clusters reveals whether, when, and how the learner used

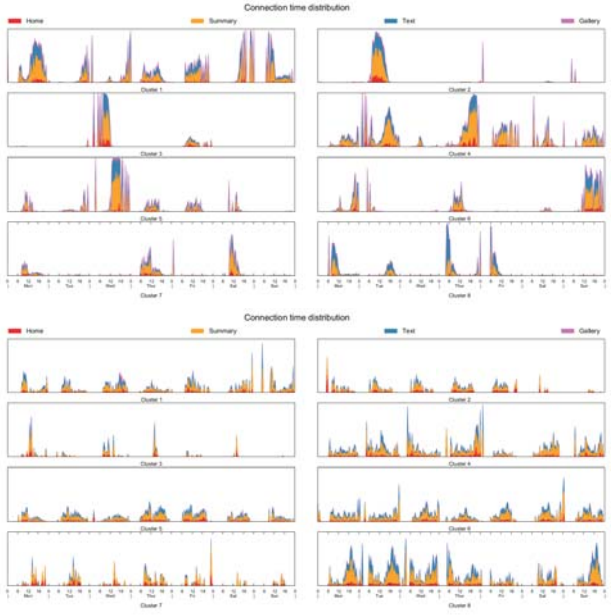


Figure 1: Resulting clusters for the session- (top) and user-based (bottom) clustering.

the materials offered by the electronic text book. Thus, the teacher can get information about the learners' navigation speed, whether part of the content was used in self-directed learning processes as expected, whether learners came up with alternative learning trajectories, and so on and so forth. This information can be used by the teacher in a formative way (cf. the concept of formative assessment, e.g., [1]), that is, it is directly used to further shape the learning process of students. For instance, in a follow-up lesson the teacher could simply draw the students attention to some parts of the book that have not or only rarely been visited. Moreover, history and learning about history could be reflected in a group discussion of learners who used the mBook resources of a particular chapter in different ways.

6. REFERENCES

[1] P. Black and D. Wiliam. Assessment and classroom learning. *Assessment in Education*, 5(1):7–74, 1998.

[2] P. Haider, L. Chiarandini, U. Brefeld, and A. Jaimes. Contextual models for user interaction on the web. In *Proc. of the Workshop on Mining and Exploiting Interpretable Local Patterns*, 2012.

[3] Y. Karagiorgi and L. Symeou. Translating constructivism into instructional design: Potential and limitations. *IFETS*, 8 (1):17–27, 2005.

[4] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[5] W. Schreiber, F. Sochatzy, and M. Ventzke. Das multimediale schulbuch - kompetenzorientiert, individualisierbar und konstruktionstransparent. In W. Schreiber, A. Schöner, and F. Sochatzy, editors, *Analyse von Schulbüchern als Grundlage empirischer Geschichtsdidaktik*, pages 212–232. Kohlhammer, 2013.

How to Visualize Success: Presenting Complex Data in a Writing Strategy Tutor

Matthew E. Jacovina, Erica L. Snow, Laura K. Allen, Rod D. Roscoe, Jennifer L. Weston, Jianmin Dai, and Danielle S. McNamara
{Matthew.Jacovina, Erica.L.Snow, LauraKAllen, Rod.Roscoe, Jennifer.Weston, Jianmin.Dai, Danielle.McNamara @asu.edu}

Arizona State University
Tempe, AZ 85287

ABSTRACT
Intelligent tutoring systems (ITSs) have been successful at improving students' performance across a variety of domains. To help achieve this widespread success, researchers have identified important behavioral and performance measures that can be used to guide instruction and feedback. Most systems, however, do not present these measures to the teachers who employ the systems in classrooms. The current paper discusses visualizations that will be displayed to teachers using the writing strategy tutor, Writing Pal. We present visualizations for both classroom and student level data and offer descriptions of each.

Keywords
Visualizations, intelligent tutoring systems, writing instruction.

1. INTRODUCTION
Over the past several decades, intelligent tutoring systems (ITSs) have been successfully developed for and implemented across a variety of domains [1]. These computer-based systems are often designed to record every interaction, behavior, and performance marker a student achieves while using the system. Research in educational data mining has used these system logs to identify what data are most predictive of overall performance and learning [2], while research in the learning sciences has used system logs to tailor instruction to individual students [3]. The synthesis of this work yields more adaptive, effective systems.

The analysis of log data has helped develop complex computational algorithms that improve adaptability within ITSs by modeling the learner [4]. Learner models can be difficult to understand without experience in modeling and educational research, and as a result, researchers have developed visualization tools to render components of these models more accessible [5]. Such tools are important because of the potential disadvantages that may emerge when the teachers who use ITSs have little understanding of their underpinnings. For instance, teachers may be less likely to use a system if they do not understand a system's feedback or what drives the feedback [6]. Moreover, if a system does not convey appropriate and timely information about students, the instructor may be unable to intervene [7].

Visualizations provide one means of aiding teachers in deciphering the complexity of ITSs and making data-driven classroom decisions [e.g., 8]. Our team is working toward providing visualizations of student progress within the Writing Pal (W-Pal), a writing strategy ITS designed for high school students. Writing Pal provides strategy instruction via lesson videos, game-based strategy practice, and essay practice with automated, formative feedback [9]. In this paper, we describe visualizations we have developed and implemented as well as those we are currently prototyping.

2. VISUALIZING DATA
Our initial goal is to provide the most relevant and understandable data to teachers through intuitive visualizations. The following sections describe visualizations that we are developing for W-Pal's *teacher interface*, where teachers view students' progress.

2.1 Classroom Level Visualizations
In a recent classroom implementation of W-Pal, five ninth grade classes with the same teacher used the system for approximately four months. We analyzed data from 90 consenting students. For the study, W-Pal's teacher interface included a spreadsheet in which teachers could track students' progress through the system activities (see Figure 1). However, during the study, this page did not provide a visual summary of the progress across students. Broadcasting the average number of activities attempted in a classroom of students who have generally stalled in their progression might prompt teachers to request that students not linger on particular topics or switch their focus. Future iterations of W-Pal will provide easily discernible bars that indicate the overall progress of classes. In Figure 1, the darker blue bars in the first four columns represent the percentage of activities attempted for those modules (a black rectangle highlights this feature).

Figure 1. Visualization of a classroom's progress in W-Pal's teacher interface; dark blue bars represent progress.

An important strength of W-Pal is the automated feedback it provides on students' essays. The teacher interface allows teachers

to view each student's submitted essays along with the feedback and score received. Currently, however, teachers do not have access to a summary of all students' performance. For example, if the majority of students are struggling to properly structure their writing in W-Pal, teachers would remain unaware until they carefully perused students' feedback messages. To provide teachers with a quickly consumable summary of the feedback that students are receiving, we are developing a visualization that displays the percentage of feedback triggered across all essays in a W-Pal class (see Figure 2). Using this information, teachers might adjust their own classroom instruction or assign students to interact with appropriate W-Pal lessons.

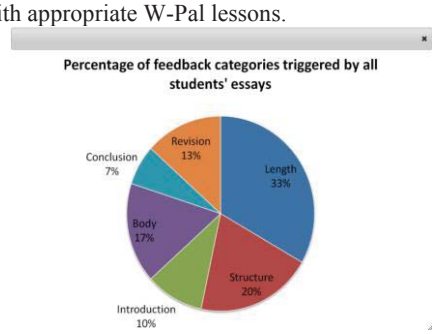


Figure 2. Visualization of the type of essay feedback students in a classroom have received.

2.2 Student Level Visualizations

Our recent classroom study also revealed that the percentage of time that students selected different activities related to their persistence in the system. For example, there was a positive correlation between the percentage of *game* activities that students selected and the number of days they used the system [$r(90) = .49$, $p < .001$]. Thus, the percentage of activities attempted (i.e., videos, games, and essay practice) could be indicative of how likely students are to persist in the system. Teachers will be presented with this information via pie charts, which are useful for visualizing proportions of a whole [10] (see Figure 3).

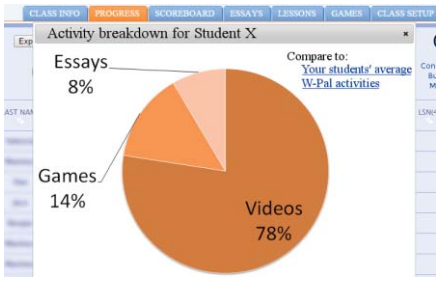


Figure 3. Visualization of the activity breakdown for an individual student.

Similar to the activity breakdown available for each student, teachers will be able click students' names in the essay window to see breakdowns of essay feedback (see Figure 2 for a similar example). If a student is struggling with writing assignments in class, this visualization will give teachers a quick view of how W-Pal has assessed areas of weakness.

3. CONCLUSION

In this paper, we argue for the importance of using visualizations to communicate data from ITSs to the teachers. Specifically, we describe classroom and student level visualizations that we are developing for the writing strategy tutor, W-Pal. When equipped with these visualizations, teachers may be more likely to use a system appropriately and to intervene when a student is not performing optimally. Future empirical work must test these visualizations, through techniques ranging from surveys to eye tracking [8], to determine their effectiveness in conveying information to teachers. As the understanding of how teachers use such visualizations grows, systems can provide teachers with intelligent tutors that better support classroom instruction.

4. ACKNOWLEDGMENTS

This work was supported by the Institute of Education Sciences (IES), USDE Grant R305A120707 to ASU. Opinions, findings, and conclusions expressed are those of the authors and do not necessarily represent views of the IES.

5. REFERENCES

[1] Graesser, A. C., McNamara, D. S., and VanLehn, K. 2005. Scaffolding deep comprehension strategies through Point & Query, AutoTutor, and iSTART. *Educational Psychologist*, 40, (2005), 225–234.

[2] Snow, E. L., Allen, L. K., Jacovina, M. E., and McNamara, D. S. 2015. Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers & Education,* 26, (2015), 378-392.

[3] Grigoriadou, M., Papanikolaou, K., Kornilakis, H., and Magoulas, G. 2001. INSPIRE: An intelligent system for personalized instruction in a remote environment. In *Proceedings of 3rd Workshop on Adaptive Hypertext and Hypermedia* (Sonthofen, Germany, July 14, 2001). Springer, Berlin, Germany, 13-24.

[4] Desmarais, M. C. and Baker, R. S. J. D. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22, (2012), 9–38.

[5] Zapata-Rivera, J. D., and Greer, J. E. 2004. Interacting with inspectable Bayesian student models. *International Journal of Artificial Intelligence in Education*, 14, (2004), 127–163.

[6] Grimes, D. and Warschauer, M. 2010. Utility in a fallible tool: A multi-site case study of automated writing evaluation. *The Journal of Technology, Learning and Assessment*, 8, (2010). Retrieved from www.jta.org.

[7] Walonoski, J. and Heffernan, N. T. 2006. Prevention of off-task gaming behavior in intelligent tutoring systems. In *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems* (Jhongli, Taiwan, June 26-30, 2006). Springer, Berlin, Germany, 722–724.

[8] Vatrapu, R., Reimann, P., Bull, S., and Johnson, M. 2013. An eye-tracking study of notational, informational, and emotional aspects of learning analytics representations. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (Leuven, Belgium, April 8-12, 2013). ACM, New York, NY, 125-134.

[9] Roscoe, R. D. and McNamara, D. S. 2013. Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105, (2013), 1010–1025.

[10] Spence, I. 2005. No humble pie: The origins and usage of a statistical chart. *Journal of Educational and Behavioral Statistics*, 30, (2005), 353–368.

Adjusting the weights of assessment elements in the evaluation of Final Year Projects

Mikel Villamañe, Mikel Larrañaga, Ainhoa Alvarez, Begoña Ferrero
Department of Languages and Computer Systems
University of the Basque Country (UPV/EHU), Spain
{mikel.v, mikel.larranaga, ainhoa.alvarez, bego.ferrero}@ehu.eus

ABSTRACT

The authors of this paper have defined a continuous evaluation methodology for Final Year Projects, in which six different evaluable items are involved. However, establishing the weights of each assessment element in the evaluation of Final Year Projects is a complex process, especially when several teachers are involved [3] like in this case. In this paper, the experiment carried out in order to establish the weight each assessment element should have in the final mark of a Final Year Project is described.

Keywords

Final Year Projects, weight adjustment, experts' validation

1. INTRODUCTION

Finishing a Final Year Project (FYP) is a challenging task for all the involved actors, either students or lecturers. In a previous work, the authors conducted a study and concluded that the main problems during projects' development are related to the evaluation process [7].

In many universities, the evaluation of the FYPs has been mainly based on a final dissertation of the work and a public oral defense in front of an examination board. This approach presents several drawbacks [6]. In order to overcome them, a set of 8 experts (teachers from the University of the Basque Country, with more than 10 years supervising FYPs) defined six elements to be taken into account and the responsible for their evaluation.

The supervisor of the project evaluates: an initial report including the project planning and requirements (*Init_Report*), the result of the design phase of the project (*Design*) and the students' attitude during the process (*Attitude).*

The evaluation board evaluates: the final report of the project (*End_Report*), the oral defense (*Defense*) and the complexity of the project (*Complexity*).

To avoid the subjectivity, an evaluation rubric was created for each of the evaluable elements [4].

2. ADJUSTING THE WEIGHTS OF THE EVALUABLES

According to the proposed FYP grading proposal [7], the final grade is computed as the weighted mean of the scores achieved in the assessable elements. Next, the experiment carried out to adjust those weights is described.

2.1 Data Set & Techniques

In order to develop a model to accurately predict the mark of a FYP, a set of graded FYPs, including the final grade provided by the evaluation board using the traditional grading way and the grades for each of the items for those projects, are required.

In this experiment, 32 FYPs were evaluated. The collected data was randomly split into two data sets, *training set*, which contained 2/3 of the collected data, and the *validation set*, entailing the remaining data.

Adjusting the weight to compute the grade as accurate as possible in relation to the grades given by the evaluation board is a regression problem. Therefore, the first technique tested was the linear regression. In this experiment the target variable is the final mark and the features are the 6 items that according to experts should influence the final mark. The objective is to determine to which extent affects each element the final mark.

During this experiment, negative coefficients were inferred (see Table 1, *LRModel)*. In the case of FYP, a negative value is not applicable as the assessable elements refer to aspects the FYP must satisfy, whilst a negative weight would mean that an undesirable or wrong feature is being evaluated. To overcome this problem, non-negativity constraints in the model should be enforced. Therefore, the Lawson-Hanson Non-negative least-squares technique [2] was used in the second phase of the experiment.

Table 1. Weights of each item in the final mark

	Weights						Analysis results	
	Init_Report	**Design**	**Attitude**	**End_Report**	**Defense**	**Complexity**	**Correlations**	**RMSE**
LRModel	0.24	0.18	-0.08	0.37	0.11	0.15	0.95	0.49
NNLSModel1	0.1	0.26	0	0.31	0.19	0.14	0.97	0.31
NNLSModel2	0.25	0.08	0.08	0.46	0.13	0	0.85	0.55
NNLSModel3	0	0	0	0.52	0.32	0.16	0.96	0.35

2.2 Validation Procedure

The validation process consisted in analyzing the extent to which the obtained model fits the data. With this objective, evaluation boards' judgments and the marks obtained using the weights of the different models were compared computing two different metrics: Pearson correlation coefficients [5] and Root-Mean-Squared Error (RMSE) [1].

The admissible error for the model has to be defined taking into account the peculiarities of the process. In this case, according to the experts, it is a common practice to round the grades to 0.5 points intervals, being very unusual to find grades not matching this criterion. For example, grades such as 7 or 7.5 were observed in the training set, whereas intermediate grades similar to 7.2 were not found. Taking this into account, for this experiment 0.5 has been asset as the maximum admissible error.

2.3 Exploratory Analysis and Working Hypothesis

The identified 6 features are considered independent factors for the final score, as they are evaluated in different stages of the FYP process. To determine the new models to compute the final grades of the FYPs, the authors stated the following hypotheses:

- **H1**: The factors identified by the expert board are appropriate predictors for the final grade of the FYPs.
- **H2**: the complexity of the FYPs is implicitly considered in the other evaluable elements.
- **H3**: The evaluation board can infer all the information needed from the *End_report* and the *Defense*.

Considering these starting hypotheses, the following models were defined for this experiment:

- **LRModel**: Model derived using linear regression and considering all the features. (Hypothesis H1)
- **NNLSModel1**: Model derived using the Lawson-Hanson Non-negative least-squares technique and considering all the features. (Hypothesis H1)
- **NNLSModel2:** Model derived using the Lawson-Hanson Non-negative least-squares technique and considering all the features except *Complexity*. (Hypotheses H1 and H2)
- **NNLSModel3**: a model derived using the Lawson-Hanson Non-negative least-squares technique only considering the *End-report*, the *Defense* and the *Complexity*. (Hypothesis H3)

3. RESULTS

In this experiment, the models described above were derived using the *training set* and tested on the *validation* set.

As it can be observed in Table 1, the linear regression technique, used for *LRModel*, led to a model with negative coefficients for some features (*Attitude*). Although the performance was remarkably good, this is not an admissible model to grade FYPs because it would mean that negative aspects of the project are being measured.

NNLSModel2 had an RMSE of 0.55 points, which did not fit in the defined admissible error range. *NNLSModel1* computed grades with 0.97 correlation with the evaluation boards' and 0.31 RMSE, whereas *NNLSModel3* achieved 0.35 RMSE.

Taking into account the calculated RMSE, the best model is *NNLSModel1* where all the features identified by the expert board are used (including *Complexity*). However, in this model *Attitude* has a weight of 0, i.e., it is not a statistically significant predictor for the final mark. Moreover, as shown in Table 1, with *NNLSModel1* an error of 0.31 in a 10-point scale has been achieved. As previously mentioned, this is an admissible error for the evaluation of FYPs because it is inferior to 0.5.

4. CONCLUSIONS AND FUTURE WORK

This paper has presented the experiment carried out in order to adjust the weights of assessment elements for the evaluation of FYPs. Several models have been evaluated, achieving a model with an error of 0.31 in a 10-point scale. One of the main results of the experiment is that the student's attitude *(Attitude)* is not statistically significant to predict the final mark.

The best performing model considers elements that must be evaluated by the supervisor of the FYP in addition to the elements assessed by the evaluation board. This suggests that, even if the evaluation board can give a grade, for a detailed evaluation, the opinion of the person who better knows the project is required.

The main future work is related to the adjustment of weights for each dimension of the rubrics. Additionally, the authors will continue validating the obtained model with new evaluations.

5. ACKNOWLEDGMENTS

This work has been supported by the Basque Government (IT722-13), the Gipuzkoa Council (FA-208/2014-B) and the University of the Basque Country (UFI11/45).

6. REFERENCES

[1] Chai, T. and Draxler, R.R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*. 7, 3 (2014), 1247–1250.

[2] Lawson, C. and Hanson, R. 1995. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics.

[3] Quevedo, J.R. and Montanes, E. 2009. Obtaining Rubric Weights for Assessments by More than One Lecturer Using a Pairwise Learning Model. (Cordoba, Spain, Jul. 2009), 289–298.

[4] Stevens, D.D., Levi, A.J. and Walvoord, B.E. 2012. *Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning*. Stylus Publishing.

[5] Taylor, R. 1990. Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*. 6, 1 (1990), 35–39.

[6] Valderrama, E., Rullan, M., Sánchez, F., Pons, J., Mans, C., Giné, F., Jiménez, L. and Peig, E. 2009. Guidelines for the final year project assessment in engineering. *39th IEEE Frontiers in Education Conference, 2009. FIE '09* (San Antonio, Texas, EE.UU., Oct. 2009), 1–5.

[7] Villamañe, M., Ferrero, B., Álvarez, A., Larrañaga, M., Arruarte, A. and Elorriaga, J.A. 2014. Dealing with common problems in engineering degrees' Final Year Projects. (Madrid, 2014), 2663–2670

Predicting students' outcome by interaction monitoring

Samara Ruiz
Department of Languages and Computer Systems
University of the Basque Country, UPV/EHU
San Sebastian, Spain
samara.ruiz@ehu.es

Maite Urretavizcaya
Department of Languages and Computer Systems
University of the Basque Country, UPV/EHU
San Sebastian, Spain
maite.urretavizcaya@ehu.es

Isabel Fernández-Castro
Department of Languages and Computer Systems
University of the Basque Country, UPV/EHU
San Sebastian, Spain
isabel.fernandez@ehu.es

ABSTRACT
In this paper we propose to predict the students' outcome by analyzing the interactions that happen in class during the course. PresenceClick lets teachers and students register their interactions during learning sessions in an agile way to give feedback in return about the students' learning progress by means of visualizations. Some of the registered interactions are the students who are attending class and a subset of the students' emotions felt during learning sessions. We have found correlations among attendance, emotions and performance in the final exam. This paper presents the study carried out to build a prediction model for the students' mark in the final exam based on these interactions. The purpose is to advice teachers about students in risk to fail.

Keywords
F2F interactions, mark prediction, linear regression, decision tree

1. INTRODUCTION
Drop out or failure is a common issue related to university students. Many studies have been carried out to detect students' problems, or even to predict the students' outcome, by applying data mining techniques to their interactions with intelligent tutoring systems [1] or course management systems [2] [3]. Other works include a wide range of potential predictors –i.e. personality factors, intelligence and aptitude tests, academic achievement, previous college achievements, or demographic data– to predict drop out and students' performance in high school [4] [5]. However, these works leave aside all the information that can be collected from the interactions that happen in face-to-face learning, the most extended way of education.

During traditional learning courses there is no way to detect problems or to know the performance of students in the final exam, except applying the teacher's intuition on the in-class students' interactions. This is even more difficult as the number of students in class grows, which is a current common issue at university worldwide. In this line, this papers aims to answer the next research questions: *Is it possible to predict the students' outcome by analyzing the interactions that happen in class? And, can we detect any interaction that especially influences the mark?*

2. PRESENCECLICK
PresenceClick is a distributed and modular environment that captures the interactions in learning sessions in an agile way. On the one hand, the *AttendanceModule* automatically captures the list of attendees to class. On the other hand, the *EmotionsModule* lets teachers capture the emotional state of the classroom related to whatever specific activity of the course. Students quantifies their emotions (six positive *–enjoyment, hope, pride, excitement, confidence* and *interest*– and six negative *–anxiety, anger, shame, hopelessness, boredom* and *frustration*–) in a 6-likert scale questionnaire based on the models described in [6] and [7]. The analyzed data belong to two subjects of Computer Science:Modular and Object Oriented Programming, (MOOP) and Basic Programming (BP). In MOOP 97 students were enrolled whereas 81 students participated in BP. The data were collected asking students to fill different event questionnaires. The MOOP students were asked three times to fill events where 41, 20 and 41 students responded respectively. The BP students were asked six times and 56, 36, 57, 48, 29 and 13 students participated (last event participation was low due to a server problem).

3. PREDICTING OUTCOME
Building a predicting model for students' outcome in the final exam was aimed to let teachers foresee those students that could be in risk to fail in the subject or even drop out.

In MOOP 44 students out of 97 enrolled attended the exam and 50 responded at least one emotion event, while 59 students attended the exam from 81 students enrolled in BP and 68 responded at least one emotion event. As the students dropping out the subject precisely are an important sample set to study, and as a considerable number of students did not attend the exam in both subjects, three different cases were studied: (Case1 - NA=F) Students non attending the exam were not considered; (Case2 - NA=T; mean=F): Students non attending the exam were assigned 0 as mark; (Case3 - NA=T; mean=T): Students non attending the exam were assigned the mean of the fails as mark, where fails are all the students with mark<5.

The three phase experiment that follows was carried out.

3.1 Phase 1: Correlation analysis
Pearson-correlation analysis was conducted between mark-attendance and mark-emotions. All the positive/negative emotions were gathered together, and the mean from all the events where each student participated was calculated in order to normalize the data. Table 1 shows the correlations for the three cases between mark-attendance, mark-positive emotions and mark-negative emotions. In both subjects *attendance and students' negative emotions influence the mark in the final exam* (except when non

attendees to exam were not considered in BP) according to literature ($p>|0.3|$) [8]. Student's positive emotions influence the mark only in MOOP. This could be due to the fact that being aware of the negative emotions is usually easier than being aware of the positive ones. In addition, we could also suppose that students expressing negative emotions in questionnaires are not lying, whereas students could increase the value of their positive emotions in order to be closer to the group feelings.

Table 1. Correlations with the mark in MOOP

	Case	Attendance	Pos emo	Neg emo
MOOP	NA=F	0.45 (p=0.0048)	0.45 (p=0.0056)	-0.46 (p=0.0034)
	NA=T, mean=F	0.6 (p=4.02e-06)	0.46 (p=0.0008)	-0.65 (p=3.78e-07)
	NA=T, mean=T	0.54 (p=4.7e-05)	0.46 (p=0.0008)	-0.59 (p=5.45e-06)
BP	NA=F	0.25 (p= 0.071)	0.13 (p= 0.35)	-0.29 (p=0.034)
	NA=T, mean=F	0.48 (p=0.0004)	0.28 (p=0.019)	-0.34 (p=0.0042)
	NA=T, mean=T	0.39 (p=0.0009)	0.23 (p=0.054)	-0.33 (p=0.006)

3.2 Phase 2: Multiple linear regression

In this stage of the experiment we looked for a model with a multiple linear regression analysis to predict the numeric mark of the student. For both subjects, 2/3 of the population was taken for training while the remaining was taken for validation. The three variables together were tested as dependent in order to predict the mark ($w+ x*$ attend.$+ y*$ posEmotions$+ z*$ negEmotions). However, for all cases the standard deviation of the model prediction error rounded two points, which implies a margin too big (in a scale grade from 0 to 10, where fails are above 5). All the emotions were also studied individually to check if any of them could explain the mark, but the error rounded the two points.

3.3 Phase 3: Classification tree

Finally, we ran a decision tree to predict whether a student drops out, fails or passes the exam. Data from both subjects were normalized and gathered in a unique dataset, and different models were tested taking into account different variables in order to find the one that better predicted the students' performance. Figure 1 presents the decision tree for the training set that best predicted the students' performance taking into account the *attendance* and the *students' negative emotions*.

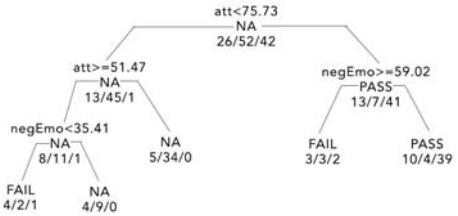


Figure 1. Training set's classification tree

As we can see in table 2 failed students are not well predicted with a 30% precision and 50% recall (F_1=37,5%), but dropping out students (F_1=86,36%) and passing students are quite well predicted (F_1=81,63%). The low correction of the fails could be due to the fact that few students are in this category and more data is required to refine the model. However, we consider that the most important measure is the recall for drop out and fail, in order to discover the students in risk and make the teacher aware. Taking into account that only 16% of failed students and 8,4% of drop out students have been predicted with PASS, we can conclude that the model is quite good, although a major sample is needed in order to adjust it for a better prediction.

Table 2. Predictions table

		Real				
		FAIL	NA	PASS	Precis.	Recall
Class	FAIL	3	3	4	30%	50%
	NA	2	19	2	82,61%	90,48%
	PASS	1	2	20	86,96%	76,92%

4. CONCLUSIONS

This paper has presented the preliminary study developed to propose a predicting model for the students' outcome in the final exam based on the interactions captured by the PresenceClick system. Those interactions data give teachers and students the possibility to avoid failure and drop out. So far, we have tested the *attendance to class* and *the students' emotions* as model predictors. The study was divided in three phases: correlation analysis, multiple linear regression and decision trees. We founded that *attendance* as well as *student's emotions* influence the mark. In particular, the *negative emotions* together with the *attendance* seem to be the interactions with bigger influence on the mark, although the multiple linear regression did not provide an accurate model. However, the decision tree brought us the possibility to foresee the students' performance in the final exam according to these factors, although a major sample is needed in order to refine the model.

5. ACKNOWLEDGMENTS

This work has been supported by the Govern of the Basque Country (IT722-13), the EHU/UPV university (PPV12/09, UFI11/45) and Gipuzkoako Foru Aldundia (FA-208/2014-B).

6. REFERENCES

[1] Baker, R., Corbett, A., Koedinger, K., 2004. Detecting Student Misuse of ITS, in: Lester, J., Vicari, R., Paraguaçu, F. (Eds.), ITS, Lecture Notes in CS, pp. 531–540.

[2] Romero, C., Ventura, S., Espejo, P.G., Hervás, C., n.d. Data mining algorithms to classify students, in: In Proc. of the 1st Int. Conf. on Educational Data Mining (EDM'08), P. 187191, 2008. 49 Data Mining 2009.

[3] Calvo-flores, M.D., Galindo, E.G., Jiménez, M.C.P., Pérez, O., n.d. 586 Current Developments in Technology-Assisted Education (2006) Predicting students ' marks from Moodle logs using neural network models.

[4] Kabakchieva Dorina, 2013. Predicting Student Performance by Using Data Mining Methods for Classification. cait 13,61.

[5] Pal, A.K., Pal, S., 2013. Data Mining Techniques in EDM for Predicting the Performance of Students. International Journal of Computer and Information 11/2013; 2(6):1110-1116.

[6] Pekrun, R., Goetz, T., Frenzel, A.C., Barchfeld, P., Perry, R.P., 2011. Measuring emotions in students' learning and performance: The AEQ. Contem. Educat. Psych. 36, 36–48.

[7] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., Christopherson, R., 2009. Emotion Sensors Go To School, in: Proc. 14th Conference on Artificial Intelligence in Education, pp. 17–24.

[8] Gray, G., McGuinness, C., Owende, P., 2013. An Investigation of Psychometric Measures for Modelling Academic Performance in Tertiary Education, in: Sixth International Conference on Educational Data Mining.

Hierarchical Dialogue Act Classification in Online Tutoring Sessions

Borhan Samei Vasile Rus Benjamin Nye Donald M. Morrison
Institute for Intelligent Systems
University of Memphis
bsamei@memphis.edu

ABSTRACT
As the corpora of online tutoring sessions grow by orders of magnitude, dialogue act classification can be used to capture increasingly fine-grained details about events during tutoring. In this paper, we apply machine learning to build models that can classify 133 (126 defined acts plus 7 to represent unknown and undefined acts) possible dialogue acts in tutorial dialog from online tutoring services. We use a data set of approximately 95000 annotated utterances to train and test our models. Each model was trained to predict top level Dialogue Acts using several learning algorithms. The best learning algorithm from top level Dialogue Acts was then applied to learn subcategories which was then applied in multi-level classification.

Keywords
Dialogue Act, Tutoring dialog, Machine Learning, Classification

1. INTRODUCTION
A speech or dialogue act is a construct in linguistics and the philosophy of language that refers to the way natural language performs actions in human-to-human language interactions, such as dialogues. [1] [6] In order to represent the Dialogue Act of an utterance, a set of Dialogue Act categories is defined. The set of categories is also known as the Dialogue Act taxonomy.

In this paper we examine different models on a relatively large data set which is extracted from one-on-one online tutoring sessions. The taxonomy used in our work is based on a hierarchical structure, i.e., each Dialogue Act has a set of sub-categories (subacts). The size of our training data is larger than the data presented in most of the previous work on Dialogue Act classification, which helps support this more fine-grained structure. We used WEKA toolkit [2] and the CRF++ package to train and test the models and Mallet [3] java library was used to train and test Logistic Regression models. Since our data is within the domain of human one-on-one tutoring sessions, this work enables further analysis of models to investigate the impact of dialog moves on learning. The feature sets used to train these models include the leading tokens of an utterance in addition to contextual information (i.e., features of previous utterances).

2. METHOD
The taxonomy used in this work was developed with the assistance of 20 subject matter experts (SMEs), all experienced tutors and tutor mentors. The resulting hierarchical taxonomy includes 15 main categories where each main dialog act category consists of different sub-categories which resulted in 133 distinct dialog acts out of which 7 categories were defined to represent unknown and undefined cases.

Once the taxonomy was available, a set of 1,438 sessions were manually tagged. The human tagging process included 4 major phases: development of taxonomy, 1st round tagging, reliability check, 2nd round tagging, reliability check, and final tagging phase.

The experts were divided into two groups: Taggers and Verifiers. In the first 2 tagging phases, each tagger was given a session transcript and asked to annotate the utterances. The resulting tagged session was then assigned to a verifier who went through the annotations, reviewed the tags and made necessary changes. In the reliability check steps, experts tagged each transcript independently.

Since the Verifiers were modifying tags already established by the Taggers in the 1st and 2nd round cases, the agreement was expected to be high. The agreement of Taggers and Verifiers was approximately 90%, with a slightly higher agreement on the second round. This shows to what extent the verifiers made changes to the initial annotations (about 10% of tags changed). The reliability checks involved completely independent tagging, in which human experts yielded an agreement of approximately 80% on top level and 60% on subact level. The final annotations were used as training data for our machine learning models. In order to build the Dialogue Act classifier, we applied the following 3 kinds of feature sets.

- Simple features: Based on previous research, 3 leading tokens of an utterance were shown to be good predictors for Dialogue Act [4]. Thus, we extracted the following features of each utterance: 1st token, 2nd token, 3rd token, last token, and length of utterance (i.e., number of tokens).

- Extended features: Using the Correlation Feature Selection (CFS) measure, we found that 1st and last token are the most predictive features and in order to add contextual information (features of prior utterances) we extended the simple features by adding the 1st and last token of three previous utterances to our feature set.

The above feature sets were used to create different models with multiple learning algorithms. Four learning algorithms were used and evaluated: Naïve Bayes, Bayesian Networks, Logistic Regression, and Conditional Random Fields (CRF). Each of the algorithms has certain properties that take into account different characteristics of data.

3. RESULTS & DISCUSSION
Based on the division of taxonomy in top-level and subcategories, we first trained and tested the models to predict the top-level Dialogue Act. Table 1 shows the results of 10-fold cross validation on the top-level classification models.

Table 1. 10-fold Cross Validation of Algorithms with Different Features for Top-level Dialogue Act Classification.

Algorithm	FeatureSet	Accuracy%	Kappa

Naïve Bayes	Simple	72.5	0.65
Naïve Bayes	Extended	72.3	0.64
Bayes Net	Simple	72.6	0.65
Bayes Net	Extended	72.5	0.65
Logistic Regression	Simple	76.6	0.70
Logistic Regression	**Extended**	**77.4**	**0.71**
CRF	Simple	72.7	0.45
CRF	Extended	71.9	0.44

As seen in table 1, the best performance on top-level classification is achieved by the Logistic Regression algorithm; however, all the algorithms yield and accuracy of more than 70%. It is interesting to note that the extended feature set does not improve the algorithms significantly which implies that adding the contextual information, i.e., prior utterances, is either not useful or not sufficiently representing the context. The diminished role of contextual features is not surprising. It has been previously indicated that they do not play a significant role in Dialogue Act classification models on a multi-party chat based tutoring system [5].

We further trained and tested models to classify utterances in the second level of Dialogue Act categories. For each Dialogue Act a classifier was trained to predict its corresponding subcategories. Table 2 shows the performance of these classifiers which were trained on 70% and tested on 30% of the dataset. A 10-fold cross-validation was not possible in this case due to too few instances for some subcategories.

Table 2. Performance of Subact Classifiers within each Dialogue Act Category using Logistic Regression algorithm.

Model	N	Accuracy%	Kappa
Answer	1130	52.8	0.43
Assertion	29890	57.6	0.42
Clarification	609	40.4	0.17
Confirmation	6620	92.6	0.77
Correction	2065	62.3	0.43
Directive	2006	61.7	0.52
Explanation	1941	54.4	0.25
Expressive	22198	76.8	0.74
Hint	341	67.6	0.34
Promise	303	95.6	0.00
Prompt	6186	64.2	0.30
Question	2553	60.7	0.49
Reminder	337	47.7	0.25
Request	14243	56.2	0.49
Suggestion	2028	70.2	0.43

As shown in Table 2, the subact classifiers yield an average accuracy of approximately 65% and kappa of 0.4. Next we created a single model to classify Dialogue Act and Subact. By combining the top-level dialogue acts with their subacts, this produced a flat taxonomy with 133 categories. Table 3 shows the performance of our models with flat taxonomy using 10-fold cross validation.

Table 3. Performance of models with flat taxonomy.

Algorithm	FeatureSet	Accuracy	Kappa
Naïve Bayes	Simple	51%	0.49
Naïve Bayes	Extended	48%	0.45
Bayes Net	**Simple**	**53%**	**0.50**
Bayes Net	Extended	51%	0.48
Logistic Regression	Extended	44%	0.42
Logistic Regression	Simple	43%	0.41

Table 3 shows that the flat taxonomy classification improved the accuracy of our model significantly when compared to the the multi-level classification. It is worth noting that these results approach the agreement of human experts when they annotated independently, which was 66%.

4. CONCLUSION

The results of the different models and algorithms showed that the top-level Dialogue Acts can be predicted with a reasonable accuracy. However to be able to tag utterances with both top-level and subcategories a combined classification needed to be applied, rather than a hierarchical approach. Multiple classification algorithm were effective, such as Naïve Bayes, Bayesian Networks, Logistic Regression, and Conditional Random Fields (CRF).

The ultimate goal of this work is to build a model to be applied to a set of not-seen and untagged data and use the Dialogue Acts as means of modeling the discourse. The proposed models in this paper can be used as initial models for a semi-supervised classifier which will ultimately identify Dialogue Acts in real time.

5. REFERENCES

[1] Austin, J. L. 1962. *How to do things with words*: Oxford.

[2] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The WEKA Data Mining Software: An Update: *SIGKDD Explor. Newsl., 11*(1), 10-18.

[3] McCallum, Andrew Kachites. 2002. "MALLET: A Machine Learning for Language Toolkit: http://mallet.cs.umass.edu.

[4] Rus, V., Moldovan, C., Niraula, N., & Graesser, A. C. 2012. Automated Discovery of Dialogue Act Categories in Educational Games: *International Educational Data Mining Society*.

[5] Samei, B., Li, H., Keshtkar, F., Rus, V., & Graesser, A. C. 2014. Context-Based Dialogue Act Classification in Intelligent Tutoring Systems: *Intelligent Tutoring Systems - 12th International Conference, {ITS} 2014, Honolulu, HI, USA, June 5-9, 2014. Proceedings*, 236-241.

[6] Searle, J. R. 1969. *Dialogue Acts: An essay in the philosophy of language*: Cambridge university press.

Towards Freshmen Performance Prediction

Hana Bydžovská
CSU and KD Lab Faculty of Informatics
Masaryk University, Brno
bydzovska@fi.muni.cz

ABSTRACT

In this paper, we deal with freshmen performance prediction. We analyze data from courses offered to students at Faculty of Informatics, Masaryk University. We supposed that the success rate of our predictions increases when we omit freshmen from our experiments as we have no study-related data about them. However, we disproved this hypothesis because there was generally no significant difference in prediction of freshmen and non-freshmen students. We also presented the attributes that were important for freshmen performance prediction.

Keywords

Student Performance, Prediction, Freshmen, Social Network Analysis, Educational Data Mining.

1. INTRODUCTION

Universities are faced with the problem of a high number of students' drop outs. Thus, researches explore what influences students' performance, and identify weak students in order to help them to improve their achievements. It is important to predict student failure as soon as possible. The task is difficult because the less data about students we have the less accurate the prediction we obtain is.

Data mining techniques represent a typical way for discovering regularity in data [3]. It allows us to build predictive models by defining valid and exact corresponding rules. Authors in [2] explored the drop-out prediction after the first year at Electrical Engineering department. Their data contained the study results of students enrolled in selected courses or the average grades gained in different groups of courses. Their results showed that decision trees belong to the most suitable algorithms. They also demonstrated that the cost-sensitive learning methods helped to bias classification errors towards preferring false positives to false negatives. Authors in [4] also investigated the prediction after the first year. They used questionnaires to get more detailed information about student habits.

We are interested in a similar problem but our task involves the prediction of student success in a course not in the whole study. Our aim is to identify the combinations of students and courses that could be predicted with a high accuracy. Due to the lack of data, we supposed that omitting freshmen (students in the first semester in their first study at the faculty) from the investigation should significantly increase the prediction accuracy. We also investigated how accurately we are able to predict the success or failure of freshmen.

2. DATA

The data used in our experiment originated from the Information System of Masaryk University. Our aim was to reveal useful attributes characterizing students in order to predict student performance in every particular course. Our data comprised of study-related and social behavior data about students. We explored the freshmen performance prediction and the observations were verified on 62 courses offered to students of the Faculty of Informatics of Masaryk University. The data sets comprised of students enrolled in courses in the years 2010-2012 and their grades. We constructed three data sets: (1) All students – 3,862 students with 42,677 grades, (2) Without freshmen – 2,927 students with 32,945 grades, (3) Only freshmen – 935 students with 9,732 grades.

2.1 Study-related data

This kind of data contained personal attributes (e.g. gender, year of birth, year of admission at the university) and data about study achievements (e.g. the number of credits to gain for enrolled, but not yet completed courses, the number of credits gained from completed courses, the number of failed courses). This data contained 42 different attributes in total.

2.2 Social Behavior Data

This kind of data described students' behavior and co-operation with other students. In order to get additional social attributes, we created sociograms. The nodes denoted users and the edges represented ties among them. The ties were calculated from the communication statistics, students' publication co-authoring, and comments among students. Particularly, we applied social network analysis methods on the sociograms to compute the values of attributes that represent the importance of each user in the network, e.g. centrality, degree, closeness, and betweenness. We also calculated the average grades of students and their friends. Finally, the social behavior data contained 131 attributes in total. We already proved that this type of data increases the accuracy of student performance prediction [1].

3. EXPERIMENT

Hypothesis. The accuracy of the student success prediction will significantly increase when we omit freshmen.

Evaluation. We utilized nine different classification algorithms implemented in Weka. We built a classifier for each investigated course because courses differ in their specialization, difficulty, and student occupancy rate. In the first place, we had to select suitable methods and compare the results of data sets with and without freshmen. We used the accuracy and coverage for comparing the results. Generally, the accuracy represents the percentage of correctly classified students. The coverage represents the amount of students for whom we can predict the success or failure.

Observations. In all cases, SMO reached the highest accuracy (with and without freshmen). We computed also baseline (the prediction into the majority class) in order to compute the percentage of successful grades. In all cases, we used 10-fold cross-validation for evaluation the results. The results comparison

can be seen in Table 1. Surprisingly, the results indicate that there is no significant improvement when we omit the freshmen. We improved the results only by 1% but for almost 10,000 grades we did not give any prediction.

Table 1. Comparison of results with and without freshmen

ALL COURSES	Accuracy		Coverage
	SMO	Baseline	
All students	80.04%	73.45%	100%
Without freshmen	81.26%	75.79%	77.2%

Naturally, the increase can be distorted by the large amount of non-freshmen students. No freshman has enrolled in 8 courses. Less than 10 freshmen were enrolled in 22 courses. Moreover, freshmen did not constitute 10% of all students in the next 18 courses. For the next investigation we selected only 14 courses where the number of freshmen is not negligible.

The results of selected 14 courses can be seen in Table 2. As can be seen, the improvement was 3.3%. However, there was a significant difference in baseline – about 7%. SMO was the most suitable method again but the results were difficult to interpret. For this reason, we also presented the accuracy using J48 for the purpose of comparison the success rate of the both approaches. We considered the J48 model to be similar enough for indication the attributes that influenced the results.

Table 2. Comparison of results for 14 courses

14 COURSES	Accuracy			Coverage
	SMO	J48	Baseline	
Without freshmen	82.07%	80.24%	77.82%	59.27%
All students	78.77%	77.48%	70.66%	100%
Only freshmen	76.56%	75.10%	67.11%	40.73%

When comparing the results presented in Table 1 and Table 2, freshmen decreased the overall accuracy in all cases. However, the difference was insignificant. The model based on J48 algorithm was explored for each course. We also investigated trees built only for the freshmen. The classifiers classified the data based on using the following attributes:

*Known study-related attributes***:** field of study, programme of the study, if the student passed the entrance test or the student was accepted without taking any entrance test, score of the entrance test, if the course is mandatory, elective, or voluntary for the student.

*Social behavior attributes***:** degree, centrality, betweenness, number of friends / average of grades of friends that already passed investigated course, number of friends / average of grades of friends that are enrolled in the course with the investigated student.

It was very interesting that the freshmen can be characterized by social attributes. They got the access to the system in June during the enrollment to their studies. During the enrollment of courses in September when we investigated their probability to pass the courses, we already had some data about their activity in the system. In order to measure the influence of the social behavior data on the freshmen performance prediction, we removed different types of data from the data set. The comparison can be seen in Table 3. SMO reached all presented results. The accuracy obtained by mining social behavior attributes was surprisingly slightly better than by mining only known study-related attributes. The best result was obtained when we used the both data types together.

Table 3. Freshmen performance prediction using different types of data

Data set	Accuracy
All attributes	76.56%
Only known study-related attributes	73.95%.
Only social behavior attributes	74.72%

Decision. The results indicated that the accuracy of the prediction was almost the same for all students regardless the status of freshmen. The freshmen passed through the similar classification paths as the non-freshmen. When we consider only the courses with a high proportion of the freshmen, the difference is higher but not significant. As a result, the hypothesis was not confirmed.

4. CONCLUSION

In this paper, we were dealing with the freshmen performance prediction. The hypothesis was that the success rate of the predictions will increase when we omit the freshmen. We disproved this hypothesis because the results sustained almost the same. The freshmen passed through the similar classification path as the non-freshmen. When we inspected the possibility of estimation only the freshmen grades, surprisingly, mining the social behavior data collected from students in the information system only in two months reached better results than mining data about results in the entrance test, course category, and the basis of the study specialization.

5. REFERENCES

[1] Bydžovská H. and Popelínský L. 2014. The Influence of Social Data on Student Success Prediction. *In Proceedings of the 18th International Database Engineering & Applications Symposium*, pp. 374-375 (2014)

[2] Dekker, G.W. and Pechenizkiy, M. and Vleeshouwers, J.M. 2009. Predicting students drop out: a case study. In T. Barnes et al. (eds.), *Proceedings of the 2nd International Conference on Educational Data Mining* (EDM'09), pp. 41-50.

[3] Marquez-Vera, C. Romero, C. and S. Ventura. 2011. Predicting school failure using data mining. *In Proceedings of the 4th International Conference on Educational Data Mining (EDM'11)*, pp. 271-276.

[4] Vandamme, J.P. and Superby, J.F. and Meskens, N. 2006. Determination of factors influencing the achievement of the first-year university students using data mining methods. In *Proceedings of the 8th international conference on intelligent tutoring systems, Educational Data Mining Workshop*, pp. 37-44.

Generalising IRT to Discriminate Between Examinees

Ahcène Boubekki
ULB Darmstadt/TU
Darmstadt/DIPF
boubekki@dipf.de

Ulf Brefeld
TU Darmstadt/DIPF
brefeld@cs.tu-darmstadt.de

Thomas Delacroix
Telecom Bretagne
thomas.delacroix@telecom-bretagne.eu

ABSTRACT

We present a generalisation of the IRT framework that allows to discriminate between examinees. Our model therefore introduces examinee parameters that can be optimised with Expectation Maximisation-like algorithms. We provide empirical results on PISA data showing that our approach leads to a more appropriate grouping of PISA countries than by test scores and socio-economic indicators.

1. INTRODUCTION

Developments in Psychometrics have led to a multitude of logistic models, ranging from simple classical test theory to sophisticated multidimensional generalizations (e.g., [2]). Usually, these generalizations focus on items and the success of solving an item depends on a particular set of skills. On the contrary, examinees are only represented by their ability although, according to the original theoretical IRT problem, items and examinees are supposed to be treated symmetrically.

In this paper, we propose to balance this asymmetry by including a discrimination parameter for examinees. We present a *homographic* parametrization that preserves symmetry and allows to derive characteristics of examinees. We report on empirical results on PISA 2012 data showing that the use of *examinee discrimination parameters* reveals insights that cannot be identified with traditional approaches.

2. A SYMMETRIC AND LOGISTIC MODEL

The traditional 1PL model [5] is given by

$$IRF_{1PL}(i,j) = \frac{1}{1+e^{\theta_i+\beta_j)}}, \tag{1}$$

where the real numbers θ and β represent the examinee's ability and the item difficulty, respectively. These parameters can be related to the score x_i and the rate of success of the question a_j by using the transformations $\beta_j = \log\left(\frac{1-a_j}{a_j}\right)$ and $\theta_i = \log\left(\frac{1-x_i}{x_i}\right)$. Note that x_i and a_j are real numbers bounded by 0 and 1. After substitution, the model can be expressed as

$$IRF_{1PL}(i,j) = \frac{a_j x_i}{a_j x_i + (1-a_j)(1-x_i)}. \tag{2}$$

A similar transformation can be applied to the 2PL [1], where $\alpha_j = b_j$ are non negative real numbers called *item discrimination*,

$$\begin{aligned} IRF_{2PL}(i,j) &= \frac{1}{1+e^{\alpha_j(\theta_i+\beta_j)}} \\ &= \frac{(a_j x_i)^{b_j}}{(a_j x_i)^{b_j} + ((1-a_j)(1-x_i))^{b_j}}. \end{aligned} \tag{3}$$

The multidimensional two-parameter logistic model (M2PL) [2] splits the items in k different skills. The examinee has an ability parameter for each skill that is affected by a skill discrimination parameter. The ability is now a vector of real numbers $\boldsymbol{\theta}_i = (\theta_{i,1}, ..., \theta_{i,k})$ and the item discrimination a vector of non-negative real numbers $\boldsymbol{\alpha}_j = (\alpha_{j,1}, ..., \alpha_{j,k})$,

$$\begin{aligned} IRF_{M2PL}(i,j) &= \frac{1}{1+e^{\boldsymbol{\alpha}_j\boldsymbol{\theta}_i} + \beta_j} \\ &= \frac{a_j \mathbf{x}_i^{b_j}}{a_j \mathbf{x}_i^{b_j} + (1-a_j)(\mathbf{1}-\mathbf{x}_i)^{b_j}}. \end{aligned} \tag{4}$$

The appealing use of *item discrimination parameters* can be translated to examinees, for instance to distinguish between a regular scholarly student and a talented, yet slacking one. Let us introduce an *examinee discrimination parameter* denoted by the non-negative real number y_i that acts as the analogue of its peer b_j. The discrimination parameters will also be decoupled from the other item or examinee parameter. This assures the identifiability of the model. The resulting model is called the *Symmetric Logistic Model* (SyLM) and given by

$$\begin{aligned} IRF_{SyLM}(i,j) &= \frac{1}{1+e^{b_j\theta_i+y_i\beta_j}} \\ &= \frac{a_j^{y_i} x_i^{b_j}}{a_j^{y_i} x_i^{b_j} + (1-a_j)^{y_i}(1-x_i)^{b_j}}. \end{aligned} \tag{5}$$

At first sight, the logistic parametrization of the SyLM appears as a special case of the M2PL by setting $\beta_j = 0$ and renaming the parameters, however, the homographic parameterization renders them intrinsically different. Actually, SyLM is closer to the 2PL as it does not subdivide items into skills although a multidimensional extension could be easily derived. For lack of space, we will thus only compare SyLM to the 1PL and 2PL.

Table 1: Synthetic results

Model	Param.	log.Lik	AIC	BIC
1PL	Log.	-3847.1	8504.3	10100.1
1PL	Hom.	-3836.6	8483.2	10079.0
2PL	Log.	-3809.2	8478.5	10172.8
2PL	Hom.	-3724.3	8308.7	10002.9
SyLM	Log.	-3809.2	9238.5	12430.1
SyLM	Hom.	-3455.5	8531.1	11722.6

3. EMPIRICAL EVALUATION

3.1 Synthetic Comparison

For each approach, logistic and homographic parameterizations are tested. Parameters are inferred by a Maximum Likelihood [4] algorithm supported by a Newton-Raphson optimization. The dataset consists of the results to the first Mathematic booklet of PISA 2012 study in France (380 examinees, 25 items). For items having two degrees of success, both cases are considered as a success. Similarly, answers entered as "not reached" or "NA" are considered as failures.

Although the results shown in Table 1 should be independent of the paramtetrization, estimations using homographic parametrizations produce better results throughout all settings. As expected, the additional parameters brought into the optimization by SyLM are crucial for the information criteria. However, comparing SyLM with the 1PL shows SyLM as the winner in two out of three cases. The decrease of the log-likelihood exceeds the increase of the AIC due to the significantly higher number of parameters.[1] The difference is even stronger for BIC and increases with the number of samples, hence naturally penalizing SyLM.

3.2 PISA Analysis

We now analyse the PISA 2012 ranking [3] and its associated country clustering with SyLM. The original grouping is based on the scores in the different tests and on social and economical variables of the countries. We focus on four pairs of countries/economies and shown in Table 2. Although Shanghai and Singapore are not reported similar, we study them together as they are the top ranked and the only ones without a similar peer. Our analysis is again performed on the Mathematics test. For each country, booklets are analyzed separately before the results are merged.

For the the twelve countries listed in Table 2, Figure 1 focuses on the distribution of *examinee's discrimination* given the *examinee's ability*. The coloring indicates the ratio of pupils having a high or a low *normalized*[2] discrimination given the fact that they have a low or a high *normalized* ability. We consider values below .25 as a low *normalized* characteristic and above .75 as a high one.

Although Switzerland and Japan are in the same PISA group, their figures are very different. The Japanese distribution is closer to the other Asiatic countries while the Swiss is similar to the German one. The geographic argument holds for Brazil and Argentina but not for USA and Russia, which are geographically and culturally very different. Again the two neighbors Canada and USA produce very different results. While the distribution for USA is closer to the British one, the Canadian one shows very different. Based on our results, an improved clustering can be proposed. Shanghai, Singapore and Japan constitute the first group; Switzerland, Germany the second. Great Britain, the USA and Russia form the third group while Brazil and Argentina make a group of their own. Canada and France remain outsiders.

[1]The 2PL counts $N+2M$ parameters, SyLM has $2N+2M$.
[2]Data is normalized by $y_i \to \frac{y_i}{1+y_i}$ and $\theta_i \to \frac{1}{1+e^{\theta_i}} = x_i$.

Table 2: PISA country grouping

QCN Shanghai	CHE Switzerland	GER Germany
SGP Singapore	JPN Japan	CAN Canada
FRA France	USA USA	BRA Brazil
GBR Great Britain	RUS Russia	ARG Argentina

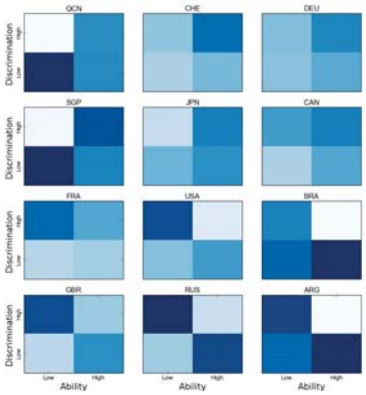


Figure 1: SyLM results for PISA

4. CONCLUSION

We proposed the Symmetric Logistic Model as a generalization of the Rasch model. Our approach can be interpreted as a symmetric 2PL at the cost of additional parameters. Empirically, our Symmetric Logistic Model showed that the PISA grouping of countries based on score and socio-economic backgrounds is suboptimal. More appropriate groups could be formed by taking examinee discrimination parameters into account.

5. REFERENCES

[1] A. Birnbaum. Some latent trait models and their use in inferring an examinee's ability. In *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968.

[2] R. L. McKinley and M. D. Reckase. An extension of the two-parameter logistic model to the multidimensional latent space. Technical report, DTIC Document, 1983.

[3] OECD. *PISA 2012 Results in Focus: What 15-year-olds know and what they can do with what they know*. OECD Publishing, 2013.

[4] N. Rose. Maximum likelihood and Bayes modal ability estimation in two-parametric IRT models: Derivations and implementation. *(Schriften zur Bildungsf.)*, 2010.

[5] N. Verhelst and C. Glas. The one parameter logistic model. In G. Fischer and I. Molenaar, editors, *Rasch Models*, pages 215–237. Springer New York, 1995.

Detection of learners with a performance inconsistent with their effort

Diego García-Saiz
Department of Software Engineering and Electronics
University of Cantabria
Avda. Los Castros s/n,
Santander, Spain
garciasad@unican.es

Marta Zorrilla
Department of Software Engineering and Electronics
University of Cantabria
Avda. Los Castros s/n,
Santander, Spain
zorrillm@unican.es

ABSTRACT

Motivation is essential to learning and performance in e-learning environments. Designing strategies to intervene in the learning process as soon as possible with the aim of keeping the learner engagement high is thus remarkably important. This paper proposes a method which allows instructors to discover learners with a performance inconsistent with the activity carried out, enabling teachers to send personalised messages to these students.

1. INTRODUCTION

Motivation is essential to carry out any kind of task successfully but, this is even more necessary for activities which require a great cognitive and time effort such as the acquisition and understanding of new knowledge to be applied suitably and rightly to problem solving. This is the case of learning processes supported by e-learning platforms where learners must adopt an active role and guide their self-learning.

To offer support and individualised help to learners, teachers need tools that help them to detect students who require advice. We, in this work, present a method which aims at detecting learners whose effort performed in the e-platform is comparable or higher than that one done by their peers but, unlike them, they do not pass the assessable assignments. These learners require a feedback different from those who are not interested in the course, thus being at risk of dropout. These feedback messages should be automatically generated by the e-learning system in order to provide students with personalized guidance, tailored to their inhomogeneous needs and requirements [1].

To our knowledge, the relationship between effort and performance has never been studied. The closest topic researched is the detection of undesirable student behaviours [3, 2] whose goal is to discover those students who have some type of problem or unusual behavior such as dropping out or academic failure. For instance, Ueno [4] proposed an animated agent which provided adaptive messages to the learners with an irregular learning process and Vellido et al. [5], characterised atypical student behaviors through robust generative relevance analysis.

Next, we describe our method and discuss the results achieved.

2. METHOD AND RESULTS

Our approach aims at detecting students who have carried out a great effort but, however, they have failed. These are thus a subset of the students that a performance classifier would classify wrongly since their activity is very similar to that performed by students who passed. Therefore our method works in two phases: first, a classifier is built in order to detect misclassified instances and next, a clustering technique is applied on the misclassified instances set of "fail" class with the aim of detecting these learners. The instances from the cluster whose weighted Euclidean distance to "fail" class prototype is the largest are our target students.

We apply our method on students' activity data from two e-learning courses hosted in Moodle with 43 and 119 learners respectively. In both, the students must carry out four assignments to pass the course. We generated two data sets, one for each course, with the activity data corresponding to the period of the first assignment(named "d1" and "d2"). The attributes used were: N# of actions performed by the student ("act"), N# of visits to the content-files ("v-re"), the SCORM resources ("v-sc"), the statistics page ("v-da"), the feedback messages provided by the instructor ("v-fe") and the html pages ("v-co"); N# of messages read ("v-fo"), posted ("a-di") and answered ("p-fo") by the student in the forum and the sum of the attributes "a-di" and "p-fo" ("pa-fo"). As class attribute, we used the mark achieved by the learner in the first assignment, pass or fail.

We configured our method for using J48 as classifier and k-means as clustering technique. The accuracy of the classifiers, evaluated with 10-CV, were 69.77% and 85.17%, with 7 and 13 instances misclassified respectively, that means, there were 7 and 13 learners who could have carried out an activity (effort) similar to those who passed the first assignment, but however they failed. To determine if these misclassified students had really a similar activity to those who passed, we performed a clustering process with these

Table 1: Clustering process on "d1"

attr.	relevance	C1	C2	Avg.
act	9	0.1835	0.4639	0.1245
v-re	4	0.093	0.438	0.1270
v-co	1	0.1017	0.3785	0.1239
v-fe	1	0	0.1667	0.0385
v-da	6	0.0435	0.3732	0.0920
a-di	2	1	0.1667	0.0769
p-fo	4	0	0.2	0.0308
pa-fo	1	0.1667	0.1944	0.0385
v-fo	2	0.3061	0.3299	0.0597
v-sc	3	0.075	0.3417	0.0952
N# ins.	-	1	6	-
dist. to avg.	-	2.0183	3.8936	-

Table 2: Clustering process on "d2"

attr.	relevance	C1	C2	C3	C4	Avg.
act	10	0.679	0.049	0.163	0.137	0.07
N# ins.	-	2	5	2	4	-
dist. to avg.	-	0.609	0.021	0,093	0,067	-

instances. Two and four clusters were created for dividing up these students. The number of clusters was manually selected by comparing the different clusters built with k ranges from 2 to 5. Next, we calculated the weighted Euclidean distance from each centroid to the mean of the well-classified instances of class "fail", being the contribution of each attribute weighted according to its relevance. Those instances which belonged to the cluster with a larger distance to the average were marked as outliers. The prototype of each cluster is shown in Tables 1 and 2. These tables also gather the relevance of each attribute ("relevance") calculated with the ClassifierSubSetEval method provided by Weka and the average value ("Avg.") of each attribute corresponding to the well-classified instances of the fail class.

As can be observed, in "d1", the cluster C1 only contains one instance which represents the activity of one of the students with the lowest activity in all course and similar to that performed by the students who failed and were well-classified. The centroid of cluster C2 is further from the average of the well-classified instances of the fail class and these, thus, are marked as outliers. In "d2", the only relevant attribute is the N# of total actions, and the instances of the cluster C1 therefore were marked as outliers.

Table 3 collects the most relevant activity performed by the six and the two students misclassified in each course respectively. In "d1", the value of most attributes is larger than the average of their class, being this difference remarkable for the attribute "act". On the one hand, the students labelled as d1s3, d1s4, d1s5 and d1s6 performed a significant activity, but failed the first assignment (q1) with a low qualification, from 0 to 4 out of 10. However, they passed the second assignment (q2) with a good mark, 9 out of 10. That means that the feedback given to them by the instructor was useful and effective, being clearly reflected the importance of giving a good feedback to the students. On the other hand, student named d1s2, even having an appreciable activity, failed the first assignment and dropped out before sending the second task. In this case, the instructor's advice was not successful. If the teacher had known the activity performed at the same time that he assessed the assignment, the message could have been written in a more motivating tone, expressly mentioning the activity already undertaken. Finally, d1s1 was detected by the method but the learner

Table 3: Students' activity

student	act	v-re	v-da	p-fo	v-sc	q1	q2
d1s1	0.23	0.30	0.24	0.00	0.19	0	0(dropout)
d1s2	0.20	0.16	0.15	0.00	0.34	3	0(dropout)
d1s3	0.91	1.00	0.72	0.60	0.63	4	9
d1s4	0.50	0.44	0.20	0.20	0.23	0	9
d1s5	0.58	0.42	0.43	0.40	0.31	3	9
d1s6	0.37	0.30	0.50	0.00	0.36	0	9
d2s1	0.84					3	8
d2s1	0.51					4.5	8.5

did not receive feedback because he did not deliver the assignment. In this case, the teacher missed the opportunity to rescue him. Regarding d2, the N# of actions performed by both students is very high in comparison with the average of the students who failed. Indeed, one of these students had a mark of 4.5 out of 10, being very close to pass. In this scenario, the feedback provided by the instructor was successful since this learner passed the second assignment with a qualification of 8.5 out of 10.

The experimentation carried out shows that our method helps to discover students whose performance do not match with the effort performed. Being able to automatically detect them would allow teachers to act quickly, sending them personalised messages oriented to keep their engagement high and avoid the dropout.

As future work, our aim is to apply this method to other virtual courses and support the teacher during the learning process in order to validate the goodness of our proposal in real online contexts. Another issue which will be addressed shortly is to evaluate the effect of using different classifiers and clustering algorithms in our proposal.

3. ACKNOWLEDGMENTS

This work has been partially financed by the PhD studentship program at University of Cantabria (Spain).

4. REFERENCES

[1] F. Castro, A. Vellido, n. Nebot, and F. Mugica. Applying data mining techniques to e-learning problems. In L. Jain, R. Tedman, and D. Tedman, editors, *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, volume 62 of *Studies in Computational Intelligence*, pages 183–221. Springer Berlin Heidelberg, 2007.

[2] A. Peña Ayala. Review: Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4):1432–1462, Mar. 2014.

[3] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(6):601–618, 2010.

[4] M. Ueno. *Data Mining in E-learning*, chapter Online outlier detection of learners' irregular learning processes, pages 261–278. Billerica,MA: WitPress, 2006.

[5] A. Vellido, F. Castro, A. Nebot, and F. Mugica. Characterization of atypical virtual campus usage behavior through robust generative relevance analysis. In *Proceedings of the 5th IASTED International Conference on Web-based Education*, WBE'06, pages 183–188, Anaheim, CA, USA, 2006. ACTA Press.

A Probabilistic Model for Knowledge Component Naming

Cyril Goutte
National Research Council
1200 Montreal Rd
Ottawa, ON, Canada
Cyril.Goutte@gmail.com

Serge Léger
National Research Council
100 rue des Aboiteaux
Moncton, NB, Canada
Serge.Leger@nrc.ca

Guillaume Durand
National Research Council
100 rue des Aboiteaux
Moncton, NB, Canada
Guillaume.Durand@nrc.ca

ABSTRACT

Recent years have seen significant advances in automatic identification of the Q-matrix necessary for cognitive diagnostic assessment. As data-driven approaches are introduced to identify latent knowledge components (KC) based on observed student performance, it becomes crucial to describe and interpret these latent KCs. We address the problem of naming knowledge components using keyword automatically extracted from item text. Our approach identifies the most discriminative keywords based on a simple probabilistic model. We show this is effective on a dataset from the PSLC datashop, outperforming baselines and retrieving unknown skill labels in nearly 50% of cases.

1. OVERVIEW

The Q-matrix, introduced by Tatsuoka [9], associates test items with attributes of students that the test intends to assess. A number of data-driven approaches were introduced to automatically identify the Q-matrix by mapping items to latent *knowledge components* (KCs), based on observed student performance [1, 6], using, e.g. matrix factorization [2, 8], clustering [5] or sparse factor analysis [4]. A crucial issue with automatic methods is that latent skills may be hard to describe and interpret. Manually-designed Q-matrices may also be insufficiently described. A data-generated description is useful in both cases.

We propose to extract *keywords* relevant to each KC from the textual content corresponding to each item. We build a simple probabilistic model, with which we score keywords. This proves surprisingly effective on a small dataset obtained from the PSLC datashop.

2. MODEL

We focus on extracting keywords from the textual content of each item (question, hints, feedback, Fig. 1). We denote by d_i the textual content (e.g. body text) of item i, and assume a Q-matrix mapping items to K skills c_k, $k = 1 \dots K$.

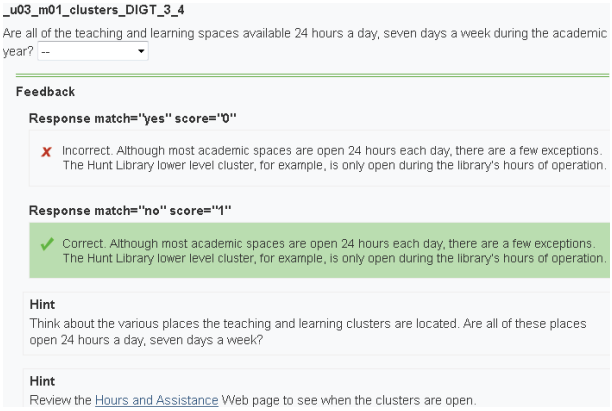


Figure 1: Example item body, feedback and hints.

These may be latent skills obtained automatically or from a manually designed Q-matrix. For eack KC we build a unigram language model estimating the relative frequency of words in each KC [7]:

$$P(w|c_k) \propto \sum_{i, d_i \in c_k} n_{wi}, \qquad \forall k \in \{1 \dots K\} \tag{1}$$

with n_{wi} the number of occurrences of word w in document d_i. $P(w|c)$ is the *profile* of c. Important words are those that are high in c's profile and low in other profiles. The symmetrized Kullback-Leibler divergence between $P(w|c)$ and the profile of all other classes, $P(w|\neg c)$, decomposes over words: $KL(c, \neg c) = \sum_w \left(P(w|c) - P(w|\neg c)\right) \log \frac{P(w|c)}{P(w|\neg c)}$. We use the contribution of each word to the KL divergence as score indicative of keywords. In order to focus on words significantly *more* frequent in c, we use the signed score:

$$\text{KL score:} \quad s_c(w) = |P(w|c) - P(w|\neg c)| \log \frac{P(w|c)}{P(w|\neg c)}. \tag{2}$$

Figure 2 illustrates this graphically. Words frequent in c but not outside (green, right) receive high positive scores. Words rare in c but frequent outside (red, left) receive negative scores. Words equally frequent in c and outside (blue) get scores close to zero: they are not specific enough.

3. EXPERIMENTAL RESULTS

We used the 100 student random sample of the "Computing@Carnegie Mellon" dataset, *OLI C@CM v2.5 - Fall 2013, Mini 1.* This OLI dataset is well suited for our study because the full text of the items is available in HTML format

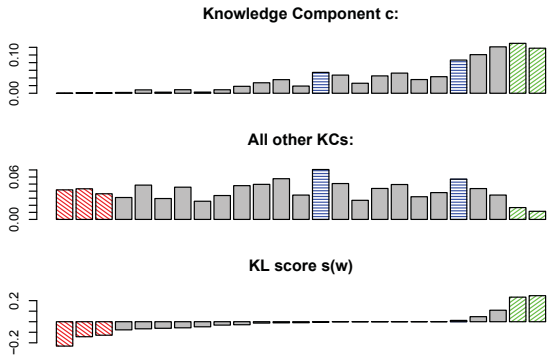


Figure 2: From KC profile, other KCs, to KL scores.

KC label	#it	Top 10 keywords (body text only)
identify-sr	52	phishing email scam social learned indicate legitimate engineering anti-phishing indicators
print	12	quota printing andrew print semester
quota		consumed printouts longer unused cost
penalties	1	maximum limitations exceed times
bandwidth		bandwidth suspended network access

Table 1: Top 10 keywords for 3 KC of various sizes.

and can be extracted. Other datasets only include screenshots. There are 912 unique steps, 31k body tokens, 11.5k hints tokens, and 41k feedback tokens, close to 84k tokens total. We pick a model in PSLC that has 108 distinct KCs with partially descriptive labels. That model assigns 1 to 52 items to each KC, for 823 items with at least 1 KC assigned. All text is tokenized, stopwords are removed, as well as tokens not containing one alphabetical character.

We estimate three different models, using Eq. (1), depending on the data considered: body text only ("body"), body and hints ("b+h"), all text ("all"). For each model, we extract up to 10 words with highest KL score (2) for each KC. Table 1 shows that even for knowledge components with very few items, the extracted keywords are clearly related to the topic suggested by the label. Although the label itself is not available when estimating the model, words from the label often appear in the keywords: this happens in 44 KCs out of 108 (41%), suggesting that the retrieved keywords are relevant. Note that some labels are vague (e.g. *identify-sr*) but the keywords provide a clear description (*phishing scams*).

We now focus on two desirable qualities for good keywords: *diversity* (keywords should differ accross KCs) and *specificity* (keywords should describe few KCs). Table 2 compares KL scores with the common strategy of picking the most frequent words (MP), using various metrics. Good descriptions should have a high number of different keywords, many of which describing a unique KC, and few KCs per keyword. The total number of keyword is fairly stable as we extract up to 10 keywords for 108 KCs. It is clear that KL extracts many more different keywords (up to 727) than MP (352 to 534). KL yields on average 1.4 (median 1) KC per keyword, whereas MP keywords describe on average 3.1 KC. There are also many more KL-generated keywords describing a unique

	total	different	unique	max
KL-body	995	**727**	**577**	**9**
KL-b+h	1005	722	558	10
KL-all	1080	639	480	19
MP-body	995	534	365	42
MP-b+h	1005	521	340	34
MP-all	1080	352	221	87

Table 2: Keyword extraction for KL vs. max. probability (MP) using text from body, b+h and all fields; total keywords, # different keywords, # with unique KC, and maximum KC per keyword.

KC. These results support the conclusion that our KL-based method provides better *diversity* and *specificity*.

Note that using more textual content (adding hints and feedback) hurts performance accross the board. We see why from the list of words describing most KCs from two methods:
KL-body: use (9) following (8) access, andrew, account (7)
MP-all: incorrect(87) correct(67) review(49) information(30)

"correct" and "incorrect" are extracted for 67 and 87 KCs, respectively, because they appear frequently in the feedback text. The KL-based approach discards them because they are equally frequent everywhere.

Acknowledgement

We used the 'OLI C@CM v2.5 - Fall 2013, Mini 1 (100 students)' dataset accessed via DataShop [3]. We thank Alida Skogsholm from CMU for her help in choosing this dataset.

4. REFERENCES

[1] T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *AAAI EDM workshop*, 2005.
[2] M. Desmarais. Mapping questions items to skills with non-negative matrix factorization. *ACM-KDD-Explorations*, 13(2), 2011.
[3] K.R. Koedinger, R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC datashop. In *Handbook of Educational Data Mining*. CRC Press, 2010.
[4] A.S. Lan, C. Studer, and R.G. Baraniuk. Quantized matrix completion for personalized learning. In *7th EDM*, 2014.
[5] N. Li, W. Cohen, and K.R. Koedinger. Discovering student models with a clustering algorithm using problem content. In *6th EDM*, 2014.
[6] J. Liu, G. Xu, and Z. Ying. Data-driven learning of Q-matrix. *Applied Psych. Measurement*, 36(7), 2012.
[7] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, 1998.
[8] Y. Sun, S. Ye, S. Inoue, and Yi Sun. Alternating recursive method for q-matrix learning. In *7th EDM*, 2014.
[9] K.K. Tatsuoka. Rule space: an approach for dealing with misconceptions based on item response theory. *J. of Educational Measurement*, 20(4), 1983.

An Improved Data-Driven Hint Selection Algorithm for Probability Tutors

Thomas W. Price
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
twprice@ncsu.edu

Collin F. Lynch
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
cflynch@ncsu.edu

Tiffany Barnes
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
tmbarnes@ncsu.edu

Min Chi
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
mchi@ncsu.edu

ABSTRACT

Data-driven systems such as the Hint Factory have been successful at providing student guidance by extracting procedural hints from prior user data. However, when only small amounts of data are available, it may be unable to do so. We present a novel hint-selection algorithm for coherent derivational domains, such as probability, which addresses this problem by searching a frontier of viable, partially matching student states. We tested this algorithm on a dataset collected from two probability tutors and performed a cold start comparison with direct state matching. We found that our algorithm provided higher value hints to students in unknown states 55.0% of the time. For some problems, it also provided higher value hints in known states.

1. INTRODUCTION

Adaptive feedback is one of the hallmarks of an Intelligent Tutoring System. This feedback often takes the form of hints, pointing a student to the next step in solving a problem. While hints can be authored by experts, more recent data-driven approaches, such as the Hint Factory [1] have shown that this feedback can be automatically generated from prior student data. The Hint Factory operates on a representation of a problem-specific dataset called an interaction network [3], where each vertex represents the state of a student's solution at some point during the problem solving process, and each edge represents a student's action. A complete solution is represented as a path from the initial state to a goal state. A new student requesting a hint is matched to a previously observed state and directed along a path to the goal state.

If too few students have been recorded, the Hint Factory is unable to match new students to existing states in the network. This is known as the *cold start problem*, a fundamental challenge in many domains. For example, when Hint Factory's original state matching algorithm was applied to BOTS, an educational programming game, a dataset of nearly 100 students provided only 40% hint coverage [4].

This paper focuses on two probability tutors in which many actions have no ordering constrains. This can produce an exponentially large state space, making the cold start problem even harder to overcome. We present a novel state matching mechanism that helps address this problem in *coherent derivational domains*. These are problem-solving domains, such as probability, physics, and logic, where: *a*) a solution S is constructed by repeated applications of domain rules to derive a goal value; *b*) taking any valid action cannot prevent the student from taking another valid action; and *c*) if S is a complete solution to the problem, then any superset of S is also a complete solution. Note that this does not prevent rule applications within a solution from having ordering constraints.

2. SELECTION ALGORITHMS

For our purposes, we assume a hint selection algorithm takes the following inputs: *a*) an interaction network, $N=(V,E)$ of previously observed states and actions; *b*) a value or ordering function $f: V \rightarrow \mathbb{R}$, which assigns "desirability" to each of the states in V; and *c*) the current state s_c of a student who is requesting a hint. In coherent derivational domains, each state $s \in V$ can be defined by the set of derived facts. Each edge $e \in E$ is annotated with an action a_e, the derivation or deletion of a fact.

Given this information, a selection algorithm attempts to find the optimal action a, such that a is a valid action in state s_c, and the value of the resulting state $f(s_a)$ is maximized. Here we derive f from the Hint Factory's value iteration procedure [1], but other functions could be used instead.

The selection algorithm employed by the Hint Factory requires that $s_c \in V$, meaning the student is in a known, or previously observed state. The algorithm then selects the successor of s_c with the highest value and returns the action which leads to this state.

In the case that s_c is unknown, meaning $s_c \notin V$, Barnes and Stamper [1] suggest using a student's previous state to generate a hint. This approach can be generalized to walking back to the last recognized state in the student's path, and using that to generate a hint. We refer to this as the "Backup Selection" algorithm.

In our selection algorithm, we first mark all $v \in V$ such that $v \subseteq s$. Beginning with the start state s_0, we traverse the graph in a depth-first fashion, following an edge e only if a_e is a deletion or derives a fact which is present in s_c. Let us call the set of states traversed in the manner T. Note that we do not *generate* states here, but explore only the previously observed states in N. We know that for any $t \in T$, $t \subseteq s_c$ and t is reachable by a known path from the start state. We define the Frontier F as the set of all states which can be reached by a single action from a state in T. A student in s_c can reach any state in the Frontier – or some superset of the Frontier state – in a single action. We then find the edge $\overrightarrow{tu}$ which maximizes $f(u)$ and return its annotated action.

3. EVALUATION

Our evaluation was based on the cold start experiment originally used to evaluate the Hint Factory [1], which was designed to measure how much data was required to provide hints to new students. Because we can always provide *some* hint by applying the Backup algorithm, we are instead interested in measuring the quality of the hints being given. Since we cannot directly measure hint quality, we will use the value function, f, described in Section 2, as an approximation of the quality. Here we use the value iteration method employed by the Hint Factory [1]. We do not make the claim this is an ideal metric, and this experiment can be easily adapted to work with any value function.

We evaluated our algorithm using combined log data from the Andes and Pyrenees probability tutors [2]. The Andes data was drawn from a prior experiment [2] and included 394 problem attempts by 66 students over 11 problems. The Pyrenees data included 999 problem attempts by 137 students on the same problem set. The tutors contain the same knowledge base, problems and solutions, allowing their data to be merged. This allowed us access to a wider variety of data than a single tutor would afford.

3.1 Procedure

For each problem, a student was selected at random and removed from the population to represent a previously unobserved student. We will call this student's path P. The remaining students who successfully solved the problem were added, one at a time and in a random order, to the network, N. Let n be the number of students added this way. After each addition, for each non-solution state in P, we calculated hints with the Backup selection algorithm and with our algorithm. We gave each of these hints a value, equal to $f(s)$, where s is the resulting state of applying the hint. In the case that this state was not in N, we used the value of the Fringe state selected by the algorithm (a superset of the resulting state). If our algorithm showed an improvement, we also recorded whether or not the state requesting a hint was known, meaning it was in N. This process was repeated 500 times to account for ordering effects.

3.2 Results

For each problem, we averaged the the percentage of *unknown* states with improved hints over all values of n. This average ranges from 33.5% to 69.8%, with an average of 55.0%. This indicates that our algorithm accomplishes its intended purpose of improving hint selection when insufficient data makes it difficult to find matching states in the network. However, while we were able to improve hints for a large *percentage* of these unknown states, the number of unknown states dropped off rapidly as n increased.

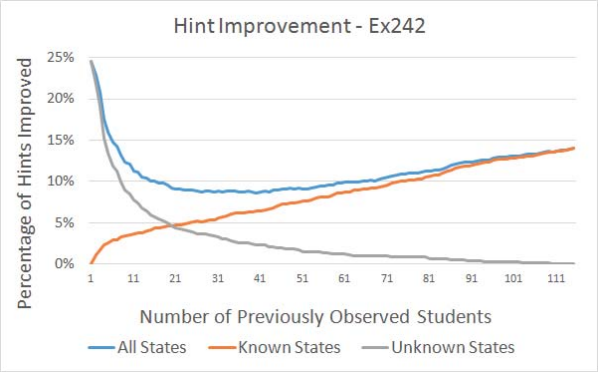


Figure 1: One cold start curve, showing the percent of hints which are improved by our algorithm (y-axis), given the number of students in N (x-axis).

For 7 of the 11 problems, our algorithm also produced improved hints for *known* states. Notably, the percentage of improved hints *increases* as more students are added to N, meaning additional data strengthens our algorithm's advantage. After all of the students were added to N, this number ranged from 3.6% to 49.7%, with an average of 17.8%. The improvement for known states seems to depend largely on the graph structure, and occurs infrequently in smaller graphs. Figure 1 depicts one cold start graph demonstrating the trends for known and unknown states.

4. CONCLUSIONS

We have presented a novel algorithm for selecting among possible data-driven hints. We have demonstrated that on average our algorithm gives a higher value hint 55.0% of the time when a student is in an unknown state, and 17.8% of the time for known states in a subset of problems.

5. ACKNOWLEDGMENTS

Work supported by NSF Grant #1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments" Min Chi & Tiffany Barnes, Co-PIs.

6. REFERENCES

[1] T. Barnes and J. Stamper. Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In *Intelligent Tutoring Systems (ITS)*, pages 373–382, 2008.

[2] M. Chi and K. VanLehn. Eliminating the gap between the high and low students through meta-cognitive strategy instruction. In *Intelligent Tutoring Systems (ITS)*, volume 5091, pages 603–613, 2008.

[3] M. Eagle and T. Barnes. Exploring Networks of Problem-Solving Interactions. In *Learning Analytics (LAK)*, 2015.

[4] B. Peddycord III, A. Hicks, and T. Barnes. Generating Hints for Programming Problems Using Intermediate Output. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, pages 92–98, 2014.

Good Communities and Bad Communities: Does membership affect performance?

Rebecca Brown
North Carolina State University
Raleigh, NC
rabrown7@ncsu.edu

Collin F. Lynch
North Carolina State University
Raleigh, NC
cflynch@ncsu.edu

Michael Eagle
North Carolina State University
Raleigh, NC
mjeagle@ncsu.edu

Jennifer Albert
North Carolina State University
Raleigh, NC
jennifer_albert@ncsu.edu

Tiffany Barnes
North Carolina State University
Raleigh, NC
tmbarnes@ncsu.edu

Ryan Baker
Teachers College, Columbia University
New York, NY
ryanshaunbaker@gmail.com

Yoav Bergner
Educational Testing Service
Princeton, NJ
ybergner@gmail.com

Danielle McNamara
Arizona State University
Phoenix, AZ
dsmcnamara1@gmail.com

Keywords
MOOC, social network, online forum, community detection

1. INTRODUCTION
The current generation of Massive Open Online Courses (MOOCs) are designed to leverage student knowledge to augment instructor guidance. Activity in these courses is typically centered on a threaded forum that, while curated by the instructors, is largely student driven. When planning MOOCs, it is commonly hoped that open forums will allow students to interact freely and that better students will help the poorer performers. It has not yet been shown, however, that this occurs in practice.

In our ongoing work, we are investigating the structure of student communities and social interactions within online and blended courses [1]. Our focus in this poster is on the structure of student communities in a MOOC and the connection between those communities and students' performance in the course. Our goal was to determine whether students in the course form strong sub-communities and whether a student's community membership is correlated with their performance. If students do form strong communities and community membership is a predictor of performance, then it would suggest either that students are forming strong relationships that help to improve their performance or that they are clustering by performance. If they do not, then it suggests that they may be able to connect freely in the forums at the expense of persistent and beneficial relationships.

2. BACKGROUND
Course-level relationships have been shown to influence students' performance and the overall success of a course. Fire et al. examined the impact of immediate peers in a traditional class and found that students' performance was significantly correlated with that of their closest peer [4]. Eckles and Stradley analyzed dropout rates and found that students with strong relationships with students who dropped out were more likely to do so themselves [3].

Rosé et al. [7] examined students' evolving social interactions in MOOCs using a Mixed-Membership Stochastic Block model which seeks to detect partially overlapping communities. They found that dropout likelihood was strongly correlated with community membership. Students who actively participated in forums early in the course were less likely to drop out later. Dawson [2] studied blended courses and found that students in the higher grade percentiles tended to have larger social networks within the course and were more likely to be connected to the instructor.

3. METHODS
Big Data in Education is a MOOC offered by Dr. Ryan Baker through the Teacher's College at Columbia University [8]. This is a 3-month long course composed of lecture videos, forum interactions, and 8 weekly assignments. All of the assignments were structured as numeric or multiple-choice exams and were graded automatically. Students were required to complete assignments within two weeks of their release and were given three attempts to do so, with the best score being used. 48,000 students enrolled in the course with 13,314 watching at least one video, 1,380 completing at least one assignment and 778 posting in the forums. Of that 778, 426 completed at least one assignment. 638 students completed the course, some managed to do so without posting in the forums.

We extracted a social network from the forums, each student, instructor, and TA was represented by a node. Each student node was annotated with their final grade. Forum users could: start new threads, add to existing threads, or add comments below existing posts. We added directed edges from the author of each item to the author of the parent post, if any, and to the authors of the items that preceded it in the current thread. We then elimi-

nated all self-loops and collapsed all parallel edges to form a simple weighted graph for analysis. We extracted two different classes of graphs. The *ALL* graphs include everyone who participated in the forums while the *Student* graphs omit the instructor and TAs. We produced two versions of each graph: one containing all participants and one that excluded students with a course grade of 0.

We identified communities using the Girvan-Newman Edge Betweenness Algorithm [5]. This algorithm takes as input an undirected graph and a desired number of communities. It operates by identifying the edge with the highest *edge-betweenness* score: the edge that sits on the shortest path between the most nodes. It then removes that edge and repeats until the desired number of disjoint graphs have been made. We applied exploratory modularity analysis to identify the *natural* number of communities [1].

Having generated the graphs and determined the natural cluster numbers, we clustered the students into communities. We treated the cluster assignment as a categorical variable and tested its correlation with final course grades. An examination of the grade distributions showed that they were non-normal, so we applied the Kruskal-Wallis (KW) test to evaluate the relationship [6]. The KW test is a non-parametric analogue of the ANOVA test.

4. RESULTS AND DISCUSSION

The raw graph contained 754 nodes and 49,896 edges. After collapsing the parallel arcs and removing self-loops we retained a total of 17,004 edges. Of the 754 nodes, 751 were students. Of those, 304 obtained a grade of 0 in the course leaving 447 nonzero students. The natural cluster number for each of the graphs is shown in Table 1 along with the result of the KW tests. As Table 1 illustrates, cluster assignment was significantly correlated with the students' grade performance for all of the graphs. A sample visualization of the student graph is shown in Figure 1.

The students formed detectable communities, and community membership was significantly correlated with performance. While the structure of the communities changed when non-students and zero-students were removed, the significance relationships held. Thus while the specific community structure is not stable under deformations, students are still most connected to others who perform at a similar level. This is consistent with prior work on traditional classrooms and issues such as dropout. It runs counter to the naïve assumption that good students will help to improve the others. While it may be the case that the better performing communities contain poorer-performing students who increased their grades through interaction with better students, the presence of so many low-grade clusters suggests that students do fragment into semi-isolated communities that do not perform very well.

More research is required to determine why these communities form, whether it is due to motivational factors or similar incoming characteristics. We present some work along these lines in [1]. We will also examine the stability of the communities over time to determine whether they can be changed or if they are a natural outgrowth of the forums and must be accepted as is.

Table 1: Community cluster numbers and Kruskal-Wallis test of student grade by community.

Users	Zeros	Clusters	K	df	p-value
All	Yes	212	349.03	211	< 0.005
All	No	173	216.15	172	< 0.02
Students	Yes	184	202.08	78	< 0.005
Students	No	169	80.93	51	< 0.005

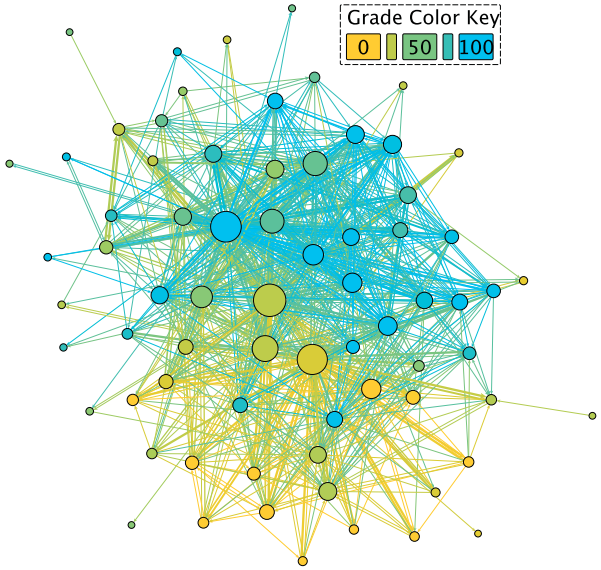


Figure 1: Student communities with edges of weight 1 removed. Nodes represent communities. Size indicates number of students. Color indicates mean grade.

5. ACKNOWLEDGMENTS

Work supported by NSF grant #1418269: "Modeling Social Interaction & Performance in STEM Learning" Yoav Bergner, Ryan Baker, Danielle S. McNamara, & Tiffany Barnes Co-PIs.

6. REFERENCES

[1] R. Brown, C. F. Lynch, Y. Wang, M. Eagle, J. Albert, T. Barnes, R. Baker, Y. Bernger, and D. McNamara. Communities of performance & communities of preference. In C. F. Lynch, T. Barnes, J. Albert, and M. Eagle, editors, *Proceedings of the 2nd International Workshop on Graph-Based Educational Data Mining*, 2015. submitted.

[2] S. Dawson. 'seeing' the learning community: An exploration of the development of a resource for monitoring online student networking. *British Journal of Educational Technology*, 41(5):736–752, 2010.

[3] J. Eckles and E. Stradley. A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2):165–180, 2012.

[4] M. Fire, G. Katz, Y. Elovici, B. Shapira, and L. Rokach. Predicting student exam's scores by analyzing social network data. In *Active Media Technology*, pages 584–595. Springer, 2012.

[5] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 99(12):7821–7826, June 2002.

[6] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

[7] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer. Social factors that contribute to attrition in moocs. In *Proc. of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM, 2014.

[8] Y. Wang, L. Paquette, and R. S. J. D. Baker. A longitudinal study on learner career advancement in moocs. *Journal of Learning Analytics*. (In Press).

A Model for Student Action Prediction in 3D Virtual Environments for Procedural Training

Diego Riofrío
ETSI Informáticos, UPM
Madrid, Spain
driofrio@fi.upm.es

Jaime Ramírez
ETSI Informáticos, UPM
Madrid, Spain
jramirez@fi.upm.es

ABSTRACT

This paper presents a predictive student action model, which uses student logs generated by a 3D virtual environment for procedural training to elaborate summarized information. This model can predict the most common behaviors by considering the sequences of more frequent actions, which is useful to anticipate common student' errors. These logs are clustered based on the number of errors made by each student and the total time that each student spent to complete the entire practice. Next, for each cluster an extended automata is created, which allows us to generate predictions more reliable to each student type. In turn, the action prediction based on this model helps an intelligent tutoring system to generate students' feedback proactively.

Keywords

Intelligent Tutoring Systems, Educational Data Mining, e-learning, procedural training, virtual environments

1. INTRODUCTION

Interactive simulations or virtual environments (VEs) have been used as tools to improve the learning by facilitating the "learning by doing" approach. Some of them show information to students through pictures, videos, interactive objects or help teachers make virtual lectures. However, there are some educative environments that can also supervise the execution of students' tasks by employing Intelligent Tutoring Systems (ITS), which provide tutoring feedback to students.

As a preamble to this work, a 3D biotechnology virtual lab was developed by our research group [4]. After evaluating this virtual lab, we saw opportunity to include the power of data mining to improve its automatic tutor by taking advantage of student logs.

Despite the work that has already been done about ITS in Educational Data Mining (EDM), the community misses more generic results [5]. Furthermore, it is also remarkable the lack of ITSs that take advantage of models developed by EDM [1].

The work presented in this paper represents a step forward towards the development of an ITS that leverages a predictive model computed by means of EDM to offer a better tutoring feedback. Moreover, this ITS is intended for procedural training in VEs and is domain independent.

Section 2 describes the proposed architecture for the ITS, which leverages the predictive student model (section 3). Finally, in section 4 we show the conclusions of this work.

2. ITS ARCHITECTURE PROPOSAL

The ITS architecture proposal is inspired on MAEVIF architecture [3], which is an extension of the ITS classical architecture for VEs.

Our main contribution resides in the Tutoring Module, which has a Tutoring Coordinator that validates the students' actions and shows error messages or hints. This module also comprises the Student Behavior Predictor (SBP) and within it lies the Predictive Student Model, which is used to find out the next most probable action from the last action made by the student. This information is used to anticipate probable students' errors, which provides a mechanism to avoid them as long as it is pedagogically appropriate.

3. PREDICTIVE STUDENT MODEL

Predictive student model uses historical data from past students and is continually refined (as Romero and Ventura recommend [5]) with actions that students under supervision are doing. In the context of the KDD Process and its adaptation into EDM formulated by Romero and Ventura [5], this model is created in Models/Patterns phase.

The model contains summarized data from historical registries of actions made by past students, and it is used to obtain the next most probable student's action. It consists of several clusters of students where each of them contains an extended automata, detailed in section 3.1. These clusters help to provide automatic tutoring adapted to each type of student. For example, if the student is committing few errors, it is more probable that his/her next action will not be an error. However,it will happen the opposite to a student who has failed more times.

The process of creation of this model is similar to the one

proposed by Bogarín et. al. [2], and it is executed at the tutor start-up. Basically, this process consists in taking events from student logs and from them data clusters of students are created based on the number of errors and the time they spent to complete the entire training process. Then, an automata for each cluster is built from the logs of the students using an incremental method. Later, at training time the SBP component updates the model with each new student's action attempt.

3.1 Extended Automata Definition

This automata consists of states (represented by circles) and transitions (represented as arrows) as shown in figure 1. Furthermore, states are grouped into three zones: Correct Flow, Irrelevant Errors and Relevant Errors Zone.

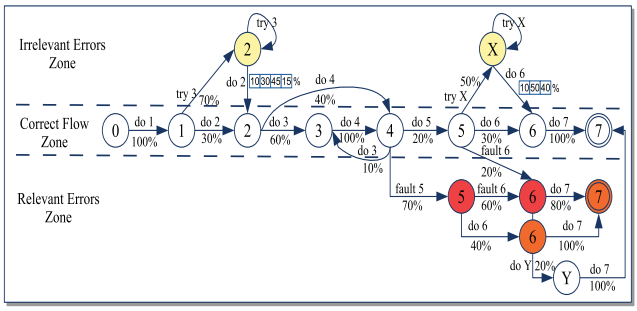


Figure 1: Example of an extended automata

Transitions denote events across an exercise such as actions or action attempts that past students have performed so far and new students may repeat in the future. An event may be a valid action of an exercise or an error detected by the tutor at the time of validating an action attempt. Accordingly, states represent the different situations derived from the events provoked by students.

Each state, and each transition, contains the number of students whose logged sequences of events have passed through, which becomes into event probabilities between states. In the case of states with loops, event frequencies to next state are reflected in a vector. In this way, the probability that a student leaves the loop on each iteration can be represented.

3.1.1 Correct Flow Zone

In this area, events represent the valid sequence of actions for an exercise, which ends up with a final satisfactory state. These states are represented by white circles.

3.1.2 Irrelevant Errors Zone

This zone groups states derived from error events that do not influence in the final result. These error events are associated with action attempts blocked by the automatic tutor (blocking errors [4]). These are graphically represented by a yellow circle.

3.1.3 Relevant Errors Zone

This area encompasses states derived from error events that actually influence in the final result, i.e. if an event of this type occurs the final result will be wrong unless a repairing action is done (non-blocking errors [4]). In this case there will be an error propagation to the subsequent states, because it does not matter what the student does later (except for some repairing action), the subsequent states will be considered also erroneous. The states derived directly from these errors are graphically represented by red circles and the subsequent correct states by orange circles.

In addition, repairing actions can be found in this area. These actions fix errors occurred earlier and redirect to one state in the correct flow.

4. CONCLUSIONS

Our proposal achieves an automatic tutoring in procedural training more adapted to each type of student by applying methods of extraction and analysis of data, which can anticipate possible errors depending on its configuration.

The principal application of the presented predictive model is to help students with preventing messages. For this, we have designed an ITS, presented above, which leverages the predictive model to provide that kind of tutoring.

We consider that the advice of an expert educator or teacher of the subject is essential at design time, despite this ITS may become very independent once its tutoring strategy is configured. This is because the resulting predictive model need to be analyzed for refining the tutoring strategy. In order to facilitate this task, it will be necessary to develop an application that displays the model to the expert or professor. In this way, he/she could visualize where students make more mistakes or where the practice is easier for them, and with this information he/she could decide where and what tutoring feedback is needed. Additionally, this could also help teacher to improve his/her own teaching.

5. ACKNOWLEDGEMENTS

Riofrío thanks Secretariat of Higher Education, Science, Technology and Innovation from Ecuador (SENESCYT).

6. REFERENCES

[1] R. S. Baker. Educational data mining: An advance for intelligent systems in education. *Intelligent Systems, IEEE*, 29(3):78–82, 2014.

[2] A. Bogarín, C. o. b. Romero, R. Cerezo, and M. S a nchez-Santill a n. Clustering for improving educational process mining. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, pages 11–15. ACM, 2014.

[3] R. Imbert, L. Sánchez, A. de Antonio, G. Méndez, and J. Ramírez. A multiagent extension for virtual reality based intelligent tutoring systems. *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 82–84, 2007.

[4] M. Rico, J. Ramirez, D. Riofrío Luzcando, M. Berrocal-Lobo, A. De Antonio, and D. Riofrio. An architecture for virtual labs in engineering education. In *Global Engineering Education Conference (EDUCON), 2012 IEEE*, pages 1–5, 2012.

[5] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

The Impact of Instructional Intervention and Practice on Help-Seeking Strategies within an ITS

Caitlin Tenison
Department of Psychology
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
ctenison@andrew.cmu.edu

Christopher J. MacLellan
Human-Computer Interaction
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
cmaclell@cs.cmu.edu

ABSTRACT

Within intelligent tutoring systems, instructional events are often embedded in the problem-solving process. As students encounter unfamiliar problems there are several actions they may take to solve it: they may explore the space by trying different actions in order to 'discover' the correct path or they can request a hint to get 'direct instruction' about how to proceed. In this paper we analyze experimental data from a tutoring system that provides two different kinds of hints: (1) interface specific hints that guide students attention to relevant portions of a worked example, supporting student discovery of next steps, and (2) procedural hints that directly tell students how to proceed. We adapted a method of sequence clustering to identify distinct hinting strategies across the two conditions. Using this method, we discovered three help-seeking strategies that change due to experimental condition and practice. We find that differences in strategy use between conditions are greatest for students that struggle to achieve mastery.

1. INTRODUCTION

As an instructional practice, tutoring supports students as they learn by doing. The tutor passively observes while the student is successful, but intervenes when the student struggles. In this paper, we explore data from two intelligent tutoring system (ITS) experimental conditions that take different approaches to assisting students. The conditions utilized adaptations of two common instructional perspectives, direct instruction and independent student discovery. These methods are often discussed in contrast to one another. Direct Instruction (DI) involves explicitly identifying and teaching the key principles, skills, and procedures for performing a specific task. The Discovery Method (DM), on the other hand, fosters a student's discovery of these principles, skills, and procedures by referring to content in the learning environment and providing indirect feedback and guidance.

To explore how DI and DM impact student learning we analyzed data from two algebra equation solving tutors [1]. In both tutors students were provided with a worked example. However, in the DI condition, students were provided with explicit procedural hints whereas in the DM condition, hints provided general information about the interface. In their initial analysis, Lee et al. looked at average actions per problems across several units and found that on some early units students in the DM tutor showed a higher proportion of mastered skills than students in the DI tutor. This effect did not persist in later units of the tutor. They concluded that, in the early units, students in the DM condition were able to learn faster with the non-verbal worked examples scaffolding than with the informative hints of the DI condition. In the current paper we aim to take a more nuanced look at how the two experimental conditions impacted help-seeking strategies and how these strategies change over the course of problem solving.

2. METHODS

The experiment was conducted within the Carnegie Learning Algebra tutor. Twenty-two high school classes were randomly assigned to the DI condition and sixteen classes were randomly assigned to the DM condition. We restricted this sample to students who had completed all experimental problems in the 'Two-step linear equation solving' unit (DI=136, DM=138). Tutors in both conditions featured a worked example that faded as students achieved mastery. In the DI condition students were provided with hints that instructed them on what procedure to do and why to do it (e.g. "To eliminate -1, add 1 to both sides of the equation because -1 +1=0"). In the DM condition students were provided with hints about how to use the interface (e.g. "Select an item from the transform menu and enter a number"). Unlike the traditional Cognitive Tutor, the initial hint was a bottom out hint. Finally, in both tutors students could make two types of mistakes, which received different feedback. If they selected off-task actions (e.g. choosing to multiply when they should have divided), they received a 'bug' telling them to undo their action and ask for a hint. If they selected an on-task action, but incorrectly applied it (e.g. dividing by an incorrect amount), they would receive 'error' feedback that their action was incorrect.

To identify distinct strategic behaviors within these tutors we first generated a matrix of all problem-solving sequences for each participant. We had a total of 5541 sequences for the DI condition and 5430 sequences for the DM condition. Correct actions were coded as 'Success', off-path actions as 'Bug', on-path actions as 'Error', and hints as 'Hint'. Next, we used a clustering method previously used to detect strategy use within an ITS [2]. This method consists of fitting a Markov Chain (MC) to each sequence, evaluating the fit of each sequence's MC to every other sequence's MC to derive a dissimilarity matrix, and using k-mediods to cluster the sequences. We found that fitting 3 clusters produced the

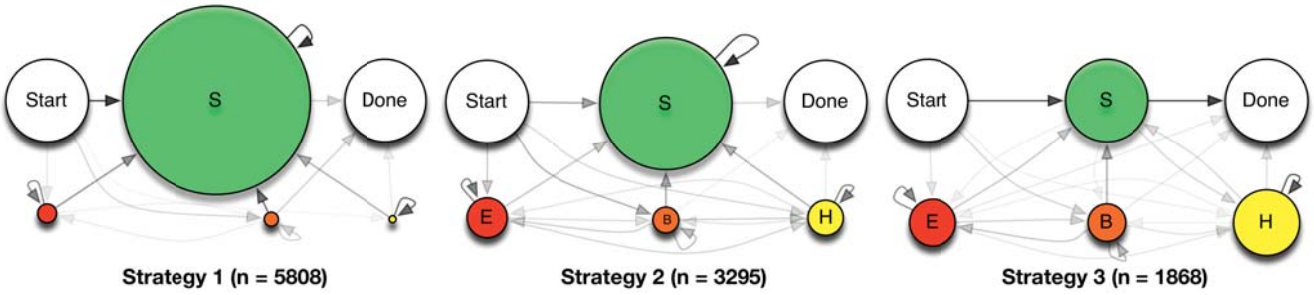


Figure 1: The student behavior for each cluster. Arrow gradients denote transition probability. Green nodes represent success, red error, orange bugs, and yellow hints.

highest average silhouette coefficient. Then, for each cluster we re-fit a single MC using all sequences assigned to that cluster to generate transition probabilities between states used to make Figure 1. After clustering the sequences we fit a binomial mixed-effects model to each cluster to better understand how students moved through the strategic clusters. Our models included fixed effects for experimental condition, the number of problems students solved (we refer to this as Practice Opportunity), and an interaction between experimental condition and practice opportunity. The models also included a random intercept for student to account for individual differences, and a random intercept for each specific problem to account for differences between the specific problems.

3. RESULTS

Figure 1 illustrates the occupancy and transitions between the different actions of the three clusters. A Chi-Squared test found that the cluster assignment of sequences from the two conditions are significantly different ($\chi^2(2) = 131.7$, $p < .001$). More sequences in the DM condition were observed in Strategy 1 (DI=2886, DI=2922) and Strategy 3 (DI=765, DM=1103) than students in the DI condition, whereas the reverse was true for Strategy 2 (DI=1890, DM=1405). Modeling Strategy 1 use, we found that the level of variability between conditions was not sufficient to include a random effect of problem. We found a marginally significant effect of intercept ($z = 1.94, p = 0.053$) along with a marginally significant interaction between the DM condition and practice opportunity ($z = 1.89, p = 0.059$). In modeling the use of Strategy 2, we found that there was a significant fixed effect of intercept ($z = -7.8, p < .001$) and of practice opportunity ($z = 3.4, p < .001$). Finally, in modeling the use of Strategy 3, we found that the random effect of practice opportunity was invariant across the different problems and model fit was improved by removing it. After removal, we found a significant fixed effect of intercept ($z = -11.2, p < .001$) as well as a significant effect of the DM condition ($z = 3.0, p < .005$). Figure 2, while not capturing the full nuanced relationship between the different factors and strategy assignments, offers some reference for understanding the model results.

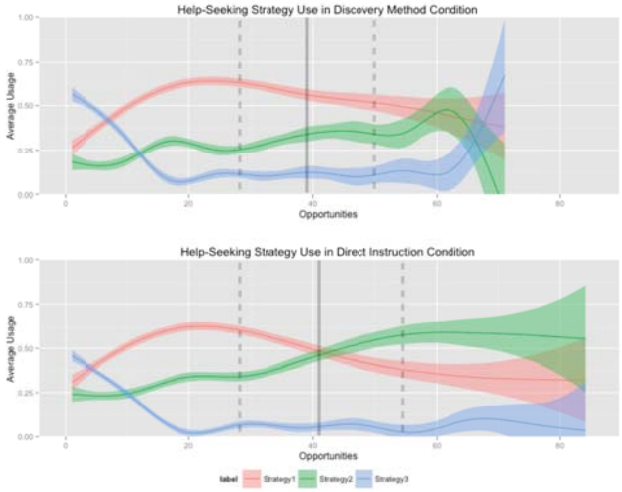


Figure 2: The average usage of strategies across practice opportunity for the two conditions. The solid vertical and dashed lines indicate the average point of mastery for DM (M=39,SD=11.5) and DI (M=41, SD 14).

In conclusion, our approach enabled us to build a picture of the strategies students use and how they change over time. Our results suggest that strategy use in the DM and DI conditions is similar, with differences appearing after higher performing students begin to reach mastery. This suggests that students who do not need help and are not exposed to the experimental manipulations have similar strategies across the two conditions. In contrast, students who achieve mastery more slowly ask for more hints, receive the manipulation, and consequently vary in their use of strategy. Future work might benefit from focusing on students that take longer to reach mastery and from coding problem type.

4. ACKNOWLEDGMENTS

This work is supported in part by IES (R305B090023). We thank Carnegie Learning, Inc., for providing the Cognitive Tutor data supporting this analysis. All opinions expressed in this article are those of the authors and do not necessarily reflect the position of the sponsoring agency.

5. REFERENCES

[1] H. S. Lee, J. R. Anderson, S. R. Berman, J. Ferris-glick, T. Nixon, and S. Ritter. Exploring optimal conditions of instructional guidence in an Algebra tutor. In *SREE Fall 2013 Conference*, 2013.

[2] C. Tenison and C. J. Maclellan. Modeling Strategy Use in an ITS : Implications for Strategic Flexibility. In *ITS*, 2014.

Predicting Performance on Dichotomous Questions: Comparing Models for Large-Scale Adaptive Testing

Jill-Jênn Vie, Fabrice Popineau, Yolaine Bourda
LRI – Bât. 650 Ada Lovelace
Université Paris-Sud
91405 Orsay, France
{jjv,popineau,bourda}@lri.fr

Jean-Bastien Grill
Inria Lille - Nord Europe
40 avenue Halley
59650 Villeneuve-d'Ascq, France
grill@clipper.ens.fr

Éric Bruillard
ENS Cachan – Bât. Cournot
61 av. du Président Wilson
94235 Cachan, France
eric.bruillard@ens-cachan.fr

ABSTRACT

Computerized adaptive testing (CAT) is a mode of testing which has gained increasing popularity over the past years. It selects the next question to ask to the examinee in order to evaluate her level efficiently, by using her answers to the previous questions. Traditionally, CAT systems have been relying on item response theory (IRT) in order to provide an effective measure of latent abilities in possibly large-scale assessments. More recently, from the perspective of providing useful feedback to examinees, other models have been studied for cognitive diagnosis. One of them is the q-matrix model, which draws a link between questions and examinee knowledge components. In this paper, we define a protocol based on performance prediction to evaluate adaptive testing algorithms. We use it to evaluate q-matrices in the context of assessments and compare their behavior to item response theory. Results computed on three real datasets of growing size and of various nature suggest that tests of different type need different models.

Keywords

Adaptive assessment, computerized adaptive testing, cognitive diagnosis, item response theory, q-matrices

1. INTRODUCTION

Automated assessment of student answers has lately gained popularity in the context of online initiatives such as massive online open courses (MOOCs). Such systems must be able to rank thousands of students for evaluation or recruiting purposes and to provide personal feedback automatically for formative purposes.

For computerized adaptive tests (CAT), item response theory (IRT) provides the most common models [3]. IRT provides a framework to evaluate the performance of individual questions, called *items*, on assessments [6]. When the intention is more formative, examinees can receive a detailed feedback, specifying which knowledge components (KCs) are mastered and which ones are not [1]. Most of these models rely on a q-matrix specifying for each question the different KCs required to solve it.

We propose a protocol to evaluate adaptive testing algorithms and use it to compare the performances of the simplest IRT model, the 1-parameter logistic one, commonly known as Rasch model, with the simplest Q-matrix model. We expect to answer the following question: given a budget of questions of a certain dataset asked according to a certain adaptive selection rule, which model performs the best at predicting the answers of the examinee over the remaining questions? We managed to get satisfactory results, enabling us to state that no model dominates in all cases: according to the type of test, either the Rasch model or the q-matrix performs the best.

2. BACKGROUND AND RELATED WORK

2.1 Item Response Theory: Rasch Model

The Rasch model estimates the latent ability of a student by a unique real number θ modeled by a random variable and characterizes each question by one real number: its difficulty d, corresponding to the ability needed to answer the question correctly. Knowing those parameters, the probability of the event "the student of ability θ answers the question of difficulty d correctly", denoted by *success*, is modeled by:

$$\Pr\{success|\theta\} = \frac{1}{1 + e^{-(\theta - d)}}.$$

The aim is first to optimize the parameters d_j for each question j and θ_i for each student i in order to fit a given train dataset. Then, throughout the test, a probability distribution over θ_i is updated after each question answered, using the Bayes' rule.

2.2 Cognitive Diagnosis Model: Q-matrix

We now present a model that tries to be more informative about the student's knowledge components. Every student is modeled by a vector of binary values $(a_1, \ldots, a_K)$, called *knowledge vector*, representing her mastery of K distinct KCs. A q-matrix Q [7] represents the different KCs involved in answering every question. In the NIDA model considered here [3], Q_{ij} is equal to 1 if the KC j is required to succeed at question i, 0 otherwise. More precisely, we denote by s_i (g_i) the *slip* (*guess*) parameter of item i. The probability of a correct response at item i is $1 - s_i$ if all KCs involved are mastered, g_i if any required KC is not mastered.

The KCs are considered independent, thus the student's knowledge vector is implemented as a vector of size K indicating for each KC the probability of the student to master it. Throughout the test, this vector is updated using Bayes' rule. From this probability distribution and with the help of our q-matrix, we can derive the probability for a given student to answer correctly any question of the test.

3. ADAPTIVE TESTING FRAMEWORK

Our student data is a dichotomous matrix of size $N_S \times N_Q$ where N_S and N_Q denote respectively the number of students and the number of questions, and c_{ij} equals 1 if student i answered the question j correctly, 0 otherwise.

We detail our random subsampling validation method. Once the model has been trained, for each student of the *test* dataset, a CAT session is simulated. In order to reduce uncertainty at most, at each step we pick the question that maximizes the Fisher information and ask it to the student. The student parameters are updated according to her answer and a performance indicator at the current step is computed. To compare it to the ground truth, we choose the negative log-likelihood [5], that we will denote by "mean error".

4. EVALUATION

We compared an R implementation of the Rasch model (IRT) and our implementation of the NIDA q-matrix model (Q) for different values of the parameter K, the number of columns of the q-matrix. Our algorithms were tested over three real datasets:

SAT dataset [4]. Results from 296 students on 40 questions from the 4 following topics of a SAT test: Mathematics, Biology, World History and French.

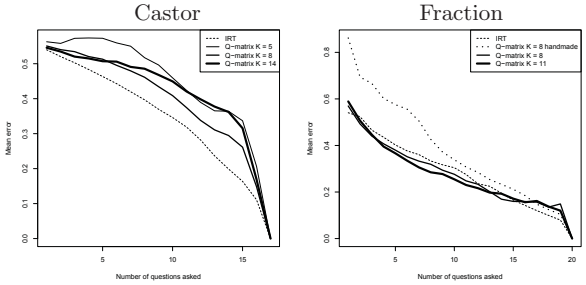
Fraction dataset [2]. Responses of 536 students to 20 questions about fraction subtraction.

Castor dataset. Answers of 6^{th} and 7^{th} graders competing in a K-12 Computer Science contest which was composed of 17 tasks. It is a 58939×17 matrix, where the (i, j) entry is 1 if contestant i got full score on task j, 0 otherwise.

Results are presented in Table 1 where the best performances are shown in bold. As a reference, 1.0 is the error obtained by the trivial algorithm affecting 1/2 to every probability. On the Castor dataset, IRT performs better than Q for any value of K throughout the whole test. On the Fraction dataset, the handmade q-matrix achieves the highest error. In the early questions of the test, Q algorithms for $K = 8$ and 11 perform slightly better than IRT. The Fraction dataset is a calculus test: it requires tangible, easy-to-define knowledge components. Therefore, after a few carefully chosen questions Q can estimate reasonably the performance of an examinee over the remaining ones. On the SAT dataset, IRT achieves the lowest error among all tested algorithms. We also observe that the variance increases throughout the test, probably because the behavior of the algorithm may vary substantially if the remaining questions are from a different topic than the beginning of the test.

5. DISCUSSION AND FUTURE WORK

Our comparison of the cognitive diagnosis model with IRT seems to indicate that q-matrices perform better on a certain type of tests; in the Fraction test, there are redundancies from one question to another in order to check that a notion is known and mastered. Conversely, IRT performs better on both the SAT test and Castor contest, which is remarkable given its simplicity. The fact that the SAT test is multidisciplinary explains the difficulty of all considered algorithms in predicting the answers, and the nature of Castor as a contest may require a notion of level instead of knowledge mastery. Therefore, in those cases, we will prefer to use the Rasch model. In order to confirm this behavior, we plan to test our implementation on many other datasets.



Castor	After 4 q.	After 10 q.	After 16 q.
Q $K = 2$	0.555 ± 0.004	0.456 ± 0.005	0.167 ± 0.012
Q $K = 5$	0.574 ± 0.004	0.460 ± 0.006	0.206 ± 0.016
Q $K = 8$	0.520 ± 0.004	0.409 ± 0.006	0.148 ± 0.013
Q $K = 11$	0.519 ± 0.004	0.462 ± 0.007	0.218 ± 0.014
Q $K = 14$	0.515 ± 0.003	0.449 ± 0.006	0.169 ± 0.014
IRT	**0.484 ± 0.003**	**0.346 ± 0.005**	**0.111 ± 0.010**
Fraction			
Q $K = 2$	0.464 ± 0.012	0.326 ± 0.013	0.196 ± 0.017
Q $K = 5$	0.440 ± 0.011	0.289 ± 0.014	**0.146 ± 0.013**
Q $K = 8$	0.407 ± 0.011	0.276 ± 0.015	0.159 ± 0.015
Q $K = 11$	**0.395 ± 0.009**	**0.255 ± 0.013**	0.156 ± 0.015
Q $K = 14$	0.422 ± 0.009	0.274 ± 0.014	0.180 ± 0.018
IRT	0.435 ± 0.012	0.304 ± 0.013	**0.142 ± 0.012**
Q* $K = 8$	0.596 ± 0.008	0.346 ± 0.007	0.182 ± 0.007
SAT			
Q $K = 2$	0.522 ± 0.007	0.417 ± 0.010	0.315 ± 0.018
Q $K = 5$	0.469 ± 0.007	0.365 ± 0.012	0.306 ± 0.019
Q $K = 8$	0.463 ± 0.007	0.367 ± 0.013	**0.242 ± 0.018**
Q $K = 11$	0.456 ± 0.008	0.364 ± 0.013	0.331 ± 0.023
Q $K = 14$	0.441 ± 0.007	0.350 ± 0.012	0.296 ± 0.021
IRT	**0.409 ± 0.008**	**0.285 ± 0.012**	**0.248 ± 0.022**

Table 1: Mean error of the different algorithms over the remaining questions of the Castor and Fraction datasets, after a certain number of questions have been asked. The dashed curve denotes the Rasch model (IRT), while the curves of growing thickness denote q-matrices (Q) of growing number of columns. The dotted curve in Fraction denotes the handmade q-matrix (Q*) [2].

6. ACKNOWLEDGEMENTS

We thank Chia-Tche Chang, Le Thanh Dung Nguyen and especially Antoine Amarilli for their valuable comments. We also thank Mathias Hiron for providing the Castor dataset. This work is supported by the Paris-Saclay Institut de la Société Numérique funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02.

7. REFERENCES

[1] Y. Cheng. When cognitive diagnosis meets computerized adaptive testing: CD-CAT. *Psychometrika*, 74(4):619–632, 2009.

[2] L. T. DeCarlo. On the analysis of fraction subtraction data: The dina model, classification, latent class sizes, and the q-matrix. *Applied Psychological Measurement*, 2010.

[3] M. C. Desmarais and R. S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[4] M. C. Desmarais et al. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In *4th International Conference on Educational Data Mining, EDM*, pages 41–50, 2011.

[5] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[6] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. *Fundamentals of item response theory*. Sage, 1991.

[7] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

The Effect of the Distribution of Predictions of User Models

Eric G. Van Inwegen Yan Wang Seth Adjei Neil Heffernan
100 Institute Rd
Worcester, MA, 01609-2280
+1-508-831-5569
{egvaninwegen, ywang14, saadjei, nth} @wpi.edu

ABSTRACT

We hypothesize that there are two basic ways that a user model can perform better than another: 1.) having test data averages that match the prediction values (we call this the *coherence* of the model) and 2.) having fewer instances near the mean prediction (we call this the *differentiation* of the model). There are several common metrics used to determine the goodness of user models; these metrics conflate coherence and differentiation. We believe that user model analyses will be improved if authors report the differentiation, as well as to include an ordering metric (e.g. AUC/A' or R^2) and an error measurement (Efron's R^2, RMSE or MAE). Lastly, we share a simplified spreadsheet that enables readers to examine these effects on their own datasets and models.

1. INTRODUCTION AND BACKGROUND

One of the goals of many in the online educational community is to more accurately predict whether a student will get the next question correct. In order to predict student responses, algorithms such as Knowledge Tracing [2], Performance Factors Analysis [6], and tabling methods [10] etc. have been developed. (See [3] for a thorough review of various user models.) Looking at only papers presented at EDM 2014, we find more than 6 new models or modifications proposed in the full papers alone [14]. Common metrics used to determine when a model is better than another include AUC/A', RMSE, MAE, and R-squared. There has been some work done (e.g. [1, 4]) looking into what metrics to use and how to interpret them [5, 11].

One can argue that current models predict the probability that a student-problem-instance (hereafter "instance") will be correct. Models such as Knowledge-Tracing ("KT"), Performance Factors Analysis ("PFA"), and their derivatives create a theoretically continuous range of predictions from 0.00 to 1.00. Even tabling models (eg. [10]) may predict a (near) continuous range of values through regressions. We argue that there are two properties of a model that will make it more accurate: 1.) How well a prediction matches the aggregate test-data, and 2.) How well the model can make predictions away from the mean.

1.1 Our Definitions

1.1.1 "Coherence"

Given a large enough data-set, we argue that an accurate model's predictions should match the test data average for a given group of instances. For example, if a model were to identify a group of instances and give that group a predicted value of 0.25, we argue that the model is most accurate when exactly one out of every four students in that condition gets the correct answer. If the model predicts 0.25, but only one out of every ten gets it right, the model's "scores" by most metrics will be improved, however, it is not as accurate as a similar model that groups that same instances together, but predicts 0.10.

1.1.2 "Differentiation"

A naive model of student knowledge might use the average score from a training dataset and predict with that probability for all instances. Arguably, more complicated user models seek to find reasons *not* to do this. The more features that a model can incorporate to move predictions away from the mean value, the better a model is at not making the mean prediction. We use the term "differentiation" in much the same way as "distribution", but do so to avoid possible confusion with the distribution of the training data.

2. METHODS

In order to visualize the impact of differentiation and coherence on the various metrics, we generate not synthetic data, but rather synthetic model outputs. To examine the effect of differentiation, a spreadsheet was created that allows the user to input prediction value, test group average, and number of instances within that group, for up to eleven groups. The spreadsheet then calculates values for AUC, A', R^2, Efron's R^2, RMSE, and MAE. A publicly shared copy of the spreadsheet can be found at: http://tinyurl.com/kznthk7. In addition to using synthetic data, the results of three models fitted to real data are explored.

3. RESULTS AND DISCUSSION

Figure 1 is a plot of the six metrics as a differentiation changes from an exceptionally steep "V" to flat to increasingly steep "A". All "models" have perfect coherence. E.g., when the model predicts 0.20, exactly 2/10 students are correct. From Figure 1, we can see that differentiation plays a role in user model "scores".

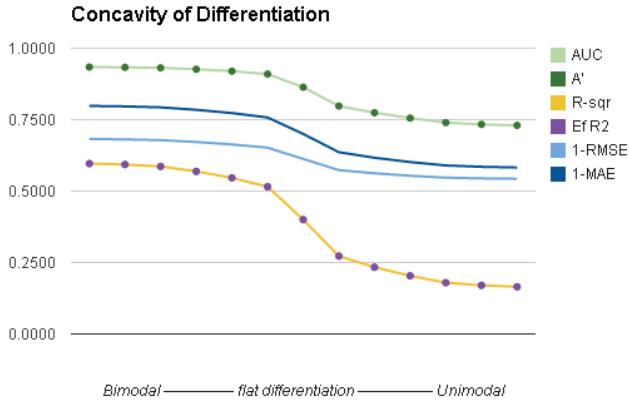


Figure 1: Chart of synthetic model outcomes showing the basic effect of changing differentiation.

To see if these ideas have merit on real data, we analyze three different models fitted to the same (~400K instance) dataset. In another paper [16], we have submitted a new user model. In that paper, the new model, called "SuperBins" (SB), is compared to Knowledge Tracing and Performance Factors Analysis, and found to be "better", according to RMSE, R^2, and AUC. If we create a frequency table of 11 groups, we will certainly lose precision, but the analysis is useful. To do so, we average the prediction values (according to their frequency) across eleven equal lengths *of prediction values* of the data set; we do the same for the test data

Table 1: A coherence-frequency table of results from three knowledge models trained and tested on the same real dataset (80/20). Model results have been averaged across 11 intervals for demonstration purposes. The prediction and test values are the weighted averages of each model within the ranges on the left.

	SB				*KT*				PFA		
Range	**pred**	**test**	**n**		*pred*	*test*	*n*		pred	test	n
0.0000 - 0.0909	**0.08**	**0.00**	**5**		*n/a*	*n/a*	*0*		0.01	0.78	9
0.0910 - 0.1818	**0.14**	**0.13**	**516**		*0.16*	*0.75*	*4*		0.13	0.53	17
0.1819 - 0.2727	**0.22**	**0.23**	**892**		*0.24*	*0.30*	*64*		0.23	0.46	56
0.2728 - 0.3636	**0.31**	**0.32**	**1829**		*0.33*	*0.28*	*704*		0.31	0.49	168
0.3637 - 0.4545	**0.41**	**0.41**	**3235**		*0.40*	*0.36*	*2565*		0.41	0.42	643
0.4546 - 0.5454	**0.50**	**0.51**	**4878**		*0.51*	*0.48*	*6978*		0.50	0.49	3539
0.5455 - 0.6363	**0.60**	**0.60**	**6355**		*0.60*	*0.61*	*8776*		0.61	0.59	7376
0.6364 - 0.7272	**0.69**	**0.69**	**9772**		*0.69*	*0.71*	*12149*		0.70	0.70	25819
0.7273 - 0.8181	**0.79**	**0.79**	**25296**		*0.78*	*0.78*	*18518*		0.77	0.78	25580
0.8182 - 0.9090	**0.86**	**0.87**	**23347**		*0.87*	*0.85*	*23600*		0.87	0.87	13811
0.9091 - 1.0000	**0.97**	**0.97**	**3074**		*0.95*	*0.95*	*5841*		0.97	0.96	2181
Metrics	**AUC** **0.728**	**R^2** **0.145**	**RMSE** **0.406**		*AUC* *0.710*	*R^2* *0.115*	*RMSE* *0.413*		AUC 0.653	R^2 0.058	RMSE 0.426
	stdev (pred): 0.166				*stdev(pred): 0.147*				stdev(pred): 0.107		

averages. E.g.., the average prediction value from 0 to 0.0909, as weighted by the frequency of each prediction was found to be 0.08 for the SuperBins model. There were no predictions in that range for KT. There were nine for PFA (eight were right), with an average prediction value of 0.01.

The analysis of coherence shows that, from 0.60 and up, all three models are reasonably accurate; i.e., the predictions closely match the test data averages. However, KT has over-predicted in the three largest of the 6 groups below 0.60. PFA appears to be reasonably consistent; however, one could argue that PFA consistently *under*-predicts in this range. Others [7] have previously reported on KT over-reporting. With this analysis, we can say that PFA has done the worst of the three at moving instances away from the mean. The major reason why SB scores so well against the other two could be its ability to bring more predictions below 0.50, while maintaining coherence.

The easiest way to measure the differentiation of the prediction values might be to report the standard deviation of prediction values. As a way to compare to the "ideal" (for that dataset), we could report either the standard deviation of the test data (0.439), or the standard deviation of the training data (0.440).

4. CONCLUSION

There are times when the metrics "scoring" user models disagree; in addition, it may be helpful for a deeper comparison.

We conclude that, if we are to accurately compare knowledge predicting models to each other, we need to look at new metrics, in addition to a mix of old metrics. We do not believe that we are proposing the "ultimate" single metric that will definitively state which model is "better". We are stating that we believe model comparison is improved when it contains (AUC or A', or R^2), and (Efron's R^2, RMSE, or MAE) and the standard deviation of the predictions. A more thorough comparison might also include coherence-frequency table analysis in an attempt to identify regions of habitual over or under prediction.

5. ACKNOWLEDGEMENTS

We would like to thank Ryan Baker and Joseph Beck for taking the time to discuss these ideas with us and make suggestions. We also acknowledge and thank funding for ASSISTments from the NSF (1316736, 1252297, 1109483, 1031398, 0742503, and 1440753), the U.S. Dept. of Ed. GAANN (P200A120238), ONR's "STEM Grand Challenges," and IES (R305A120125, R305C100024).

6. REFERENCES

[1] Beck, J. E., & Xiong, X. (2013). Limits to accuracy: How well can we do at student modeling. *Educational Data Mining*.

[2] Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4(4), 253-278.

[3] Desmarais, M. C., & d Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, *22*(1-2), 9-38.

[4] Dhanani, A., Lee, S. Y., Phothilimthana, P., & Pardos, Z. (2014). A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley.

[5] Fogarty, J., Baker, R. S., & Hudson, S. E. (2005). Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. *Proceedings of Graphics Interface 2005*. Canadian Human-Computer Communications Society.

[6] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.

[7] Qiu, Y., Pardos, Z. & Heffernan, N. (2012). Towards data driven user model improvement. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Florida Artificial Intelligence Research Society (FLAIRS 2012). pp. 462-465.

[8] Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) Proceedings of the 7th International Conference on Educational Data Mining.

[9] Van Inwegen, E. G., Adjei, S. A., Wang, Y., & Heffernan, N. T. "Using Partial Credit and Response History to Model User Knowledge" *accepted into Educational Data Mining 2015*.

[10] Wang, Y., & Heffernan, N. T. (2011). The" Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *FLAIRS Conference*.

[11] Yudelson, M., Pavlik Jr, P. I., & Koedinger, K. R. (2011). User Modeling–A Notoriously Black Art. *User Modeling, Adaption and Personalization*, 317-328.

Predicting Student Aptitude Using Performance History

Anthony F. Botelho
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609-2280
abotelho@wpi.edu

Seth A. Adjei
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609-2280
saadjei@wpi.edu

Hao Wan
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609-2280
hale@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609-2280
nth@wpi.edu

ABSTRACT

Many tutoring systems currently in use provide a wealth of information pertaining to student learning over long periods of time. Providing meaningful representations of student performance can indicate levels of knowledge and understanding that can alert instructors to potential struggling students in order to provide aid where it is needed; it is the goal of many researchers to even provide such indication preemptively in order to intervene before students become frustrated when attempting new skills. The goal of this work is to utilize student performance history to provide a means of quantizing student aptitude, defined here as the speed at which a student learns, and then using this measurement to predict the speed at which each student will learn the next skill before beginning. Observing a dataset of 21 skills, we compare two methods of predicting aptitude to majority class predictions at the skill level. Our results illustrate how our proposed methods exhibit different strengths in predicting student aptitude when compared to majority class, and may be used to direct attention to a struggling student before attempting a new skill.

Keywords

Aptitude, Student Knowledge, Intelligent Tutoring Systems

1. INTRODUCTION

Many instructors rely on intelligent tutoring systems (ITS) as a means of extending student learning outside the classroom. Many such systems, such as the ASSISTments system used in this work, provide a wealth of student performance data that is often underutilized. While many systems have focused on and have shown success in predicting next problem correctness, such information is only useful to instructors in a short time-span as students are completing assignments. Furthermore, many of these models rely on latent variables that leas to problems of identifiability [1] when attempting to draw conclusions of student knowledge.

The purpose of this work is to observe and predict student learning rates, referenced throughout this paper as aptitude; this value is expressed as a metric in terms of completion speed (cs), or the number of problems a student needs to complete the assignment (described further in the next section). Such a measure of aptitude in prerequisite skills has shown to be successful in predicting initial knowledge, represented as correctness, on a subsequent skill [2], illustrating that the two concepts are related, but from that work, it is unclear as to whether student aptitude is transitive across skills. In this work, therefore, we strive to answer the following research questions:

1. Do students exhibit similar degrees of aptitude across skills?
2. Are changes in student aptitude across skills predictable?
3. Can a student's aptitude in previous skills be used to construct a reliable prediction of completion speed in a new skill before it is begun?

2. METHODOLOGY

The dataset[1] used in this work is comprised of real-world data from PLACEments test data reported from the ASSISTments tutoring system. Data pertaining to 21 unique observable skills was extracted. Here, we define a skill as observable if it contains data from more than 10 unique students, and no less than half of the students must have completed the skill. ASSISTments defines skill completion in terms of 3 consecutive correct answers.

We used a simple binning method implemented in similar research [2][3] to place students into one of five categories based on completion speed in order to represent different levels of aptitude. As aptitude is an independent concept of domain knowledge, a student's entire recorded performance history, regardless of the prerequisite structure, was used to categorize each student. Observing each student's performance over several skills, we used a moving average of student completion rates of each skill ordered from oldest

[1]The original raw dataset can be found at the following link: http://bit.ly/1DVbHdB.

to most recent. Equation 1 displays the formula for this method. For our implementation, we used a value of 0.3 for alpha.

$$A_t = ((1 - \alpha) * A_{t-1}) + (\alpha * V_t) \quad (1)$$

Table 1: The ranges of completion speed represented by each bin with corresponding the quantized aptitude value.

Bin Number	Completion Speed(cs)	Quantized Value
1	$3 \leq$ cs ≤ 4	1
2	$4 <$ cs < 8	0.75
3	$8 \leq$ cs	0.5
4	DNF, pcor $\geq .667$	0.25
5	DNF, pcor $< .667$	0

Once an average completion speed, in terms of number of problems needed to reach three sequential correct responses, each student is placed in the corresponding bin described in Table 1. Bins 4 and 5 contain students that did not finish (DNF) at least one previous skill, and are instead split based on the average percent correctness (pcor) across all previous skills. The quantized values are chosen arbitrarily to discretize the learning rate that is intended to be represented by each bin.

2.1 Experiments

Our first prediction method, referenced as Same Bin Prediction (SBP) in our results section, simply uses the average completion speed of each student's performance history to determine in which bin to place each student. The method then simply uses that bin's quantized value as a prediction for the new skill. Both the SBP and majority class are then compared to each student's actual completion speed, expressed as a quantized bin value, to determine both error rates.

Our second experiment attempts to make predictions again using each student's performance history, but by also taking into account changes in aptitude across skills. Our first experiment assumes that most students will exhibit the same level of aptitude in a new skill as in previous skills. This experiment takes into account the realization that differences in skill difficulty may cause fluctuations in our aptitude measurements. Our second method, referenced as Transitioning Bin Prediction (TBP) in our results section, builds off of the previous SBP prediction by calculating an offset transition value. For example, if half the students in bin 1 (value = 1) remained in that bin for the new skill, while half transitioned to bin 2 (value = 0.75), an offset value of -0.125 would be applied to all predictions of bin 1. A negative offset indicates that many students required more opportunities to complete than normal, while a positive offset indicates the reverse. The prediction is normalized to a value between 0 and 1 to make full use of our quantized values

3. RESULTS AND CONCLUSIONS

Table 2 contains the RMSE results of each prediction method divided by each bin of the new skill. The success of the majority class predictions extends across higher aptitude students, while the TBP method provides the most accurate predictions over students in the lower aptitude bins.

Table 2: Average RMSE of the skill level analysis divided by bin.

Bin of New Skill	Majority Class	SBP	TBP
1	0.230	0.498	0.358
2	0.120	0.356	0.170
3	0.284	0.362	0.205
4	0.307	0.526	0.251
5	0.571	0.659	0.497

Table 3: Percent correctness at the skill level divided by bin.

Bin of New Skill	Majority Class	SBP	TBP
1	0.709	0.479	0.500
2	0.280	0.245	0.268
3	0.102	0.251	0.200
4	0	0.029	0.129
5	0	0.041	0.333

Each method described in this work exhibited different strengths, including the simple majority class predictions. It is often for the benefit of both teachers and students that a model represent meaningful information beyond the provision of predictive accuracy. The SBP method, for example, while not excelling in any one category, illustrates tendencies of aptitude mobility. Such methods may act as a means of better understanding and developing course structure and skill relationships.

The fact that the proposed prediction methods fail to outperform majority class overall suggests that using all performance history is not by itself a strong predictor of future performance, and is instead dependent to some degree on skill-based attributes. This work ignores prerequisite skill hierarchies available in many tutoring systems and MOOCs, using all previous performance history. Using prerequisite data may lead to stronger predictions, or at the very least provide indications of strong and weak skill relationships. Knowing more information about such skill relationships could provide better indications of when performance history is most useful as a predictor.

4. ACKNOWLEDGMENTS

We acknowledge funding for ASSISTments from the NSF (1316736, 1252297, 1109483, 1031398, 0742503, 1440753), the U.S. Dept. of Ed. GAANN (P200A120238), ONR's "STEM Grand Challenges," and IES (R305A120125, R305C100024).

5. REFERENCES

[1] J. E. Beck and K. min Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146. Springer Berlin Heidelberg, 2007.

[2] A. Botelho, H. Wan, and N. Heffernan. The prediction of student first response using prerequisite skills. In *Learning at Scale*, 2015.

[3] X. Xiong, S. Li, and J. E. Beck. Will you get it right next week: Predict delayed performance in enhanced its mastery cycle. In *FLAIRS Conference*, 2013.

Discovering Concept Maps from Textual Sources

R.P. Jagadeesh Chandra Bose Om Deshmukh B. Ravindra

Xerox Research Center India
Etamin Block 3, 4th Floor, Wing-A, Prestige Tech Park II, Bangalore, India 560103.

{jagadeesh.prabhakara, om.deshmukh}@xerox.com

ABSTRACT

Concept maps and knowledge maps, often used as learning materials, enable users to recognize important concepts and the relationships between them. For example, concept maps can be used to provide adaptive learning guidance for learners such as path systems for curriculum sequencing to improve the effectiveness of learning process. Generation of concept maps typically involve domain experts, which makes it costly. In this paper, we propose a framework for discovering concepts and their relationships (such as prerequisites and relatedness) by analyzing content from textual sources such as a textbook. We present a prototype implementation of the framework and show that meaningful relationships can be uncovered.

1. INTRODUCTION

In any given learning setting, a hierarchy of concepts (set by experts) is provided and the learner is expected to follow through these concepts in the specified order, e.g., Table of Contents (ToC), which indicates that concepts appearing in earlier chapters *are* (sometimes *'may be'*) pre-requisites for the concepts discussed in the later chapters. Similarly, end-of-the-book index indicates prominent occurrences of the main concepts (and some relationships between them) discussed in the book. In both the cases, the relationship is static, is designed by the experts and is restricted to the pre-populated list of concepts. As we move towards personalized learning, such a knowledge-driven static elicitation is inadequate. e.g., if the immediate goal of the learner is to understand concepts in chapter L, s/he may only have to go through a select 'n' sections of some chapters till L. Consider another example, if a learner has to know which concepts co-occur or which concepts predominantly occur before a particular concept C and are relevant to the concept C. This information is not easily available either from the ToC or from the "end-of-the-book index".

Concept map is a knowledge visualization tool that represents concepts and relationships between them as a graph. Nodes in the graph correspond to concepts and edges depict the relationship between concepts. In recent years, concept maps are widely used for facilitating meaningful learning, capturing and archiving expert knowledge, and organizing and navigating large volumes of information. In adaptive learning, concept maps can be used to give learning guidance by demonstrating how the learning status of a concept can possibly be influenced by learning status of other concepts [3]. Construction of concept maps is a complex task and typically requires manual effort of domain experts, which is costly and time consuming.

In this paper, we propose a framework for automatic generation of concept maps from textual sources such as a textbook and course webpages. We discover concepts by exploiting the structural information such as table of contents and font information and establish how closely two concepts are related to each other where the relation is defined on how strongly one concept is being referred to/discussed in another. The proposed approach is implemented and applied on several subjects. Our initial results indicate that we are able to discover meaningful relationships.

The remainder of this paper is organized as follows. Related work is presented in Section 2. We discuss our approach of discovering concept maps in Section 3. Section 4 presents some experimental results. Section 5 concludes with some directions for future work.

2. RELATED WORK

Concept map mining refers to the automatic or semi-automatic creation of concept maps from documents [4]. Concept map mining can be broadly divided into two stages: (i) concept identification and (ii) concept relationships association. Concept identification is typically done using dictionaries or statistical means (e.g., frequent words). Relation between concepts is typically defined over word-cooccurrences. In our work, we do not use any dictionary of terms. Instead, we rely on structural information such as bookmarks, table of contents, and font information manifested in data sources to discover concepts. Furthemore, when discovering relationships, we not only look at co-occurence of concepts within a sentence but scope it to larger segments such as a section and chapter.

3. GENERATION OF CONCEPT MAPS

Concept maps should provide support for modular nature of the subject matter and the interconnections between knowledge modules (concepts). Formally, a concept map can be defined as a tuple $< C, R, L >$ where $C = \{c_1, c_2, \ldots, c_n\}$ is a set of concepts; $L = \{l_1, l_2, \ldots, l_k\}$ is a set of labels. $R = \{r_1, r_2, \ldots, r_m\} \subseteq C \times C \times L$ is a set of relationships among concepts. Each relation $r_j = (c_p, c_q, l_s) \in R, p \neq q, 1 \leq p, q \leq n, 1 \leq j \leq m, 1 \leq s \leq k$ defines a relation-

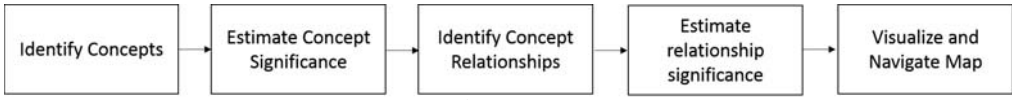


Figure 1: Approach Overview

ship between concept c_p and c_q which is labeled l_s. Optionally each relation r_j can also be associated with a weight $w_j \in \Re^+$. Figure 1 presents an overview of our approach and is comprised of five steps:

1. Identify Concepts: We exploit structural and font information such as bookmarks, table of contents, and index (glossary) in e-textbooks, and headers and font information in html pages for this step. Text processing such as tokenizing, stemming, and stop word removal are then applied. Concepts are identified as either individual words or n-words ($n > 1$)

2. Estimate Concept Significance: We estimate the significance of concepts automatically using different criteria: (i) frequency of occurrence (frequent concepts are more significant than infrequent ones) (ii) importance of a concept w.r.t the examinations/evaluations and (iii) font related information (larger font concepts are more significant than smaller fonts). The three criteria mentioned above can be grouped together using weights.

3. Identify Concept Relationships: Several types of relationships can be defined among concepts, e.g., superclass-subclass (one concept is *more general* than another), prerequisite relation (a concept A is said to be a pre-requisite for concept B), etc. The table of contents in a document directly gives a (partial) hierarchical structure among concepts. Apart from the hierarchical relationship, concepts can also be horizontally related e.g., *relevant to* and *mentioned by* as discussed in [1]. We consider the *mentioned by* relation, which is used to express the fact that two concepts are related of the type A *refers-to* B, A *discusses* B, A *mentions* B. Note that *mentioned by* is an *asymmetric* and *not necessarily transitive* relation.

4. Estimate Relationship Significance: Relationship significance is estimated using *term co-occurrence* as a basis. For each concept, in the pages where it manifests, we also estimate which other concepts manifest in those pages and how often do they manifest. The degree of relatedness is obtained by the frequency at which the concept is used, e.g., if concept c_j manifests f_j times when describing concept c_i and if f_i is the frequency of occurrence of concept c_i, then the weight of the edge between c_j and c_i can be defined as f_j/f_i. We also consider normalized weights.

5. Visualize and Navigate Map: The concepts and their relationships can be visualized as a graph $G = (V, E)$ where V, the set of vertices, correspond to the concepts and E, the set of edges, correspond to the relationship between concepts. Nodes and edges can be annotated to provide rich information and enable the navigation of these maps e.g., size of the node can be used to depict the significance of a concept, color of the node can be used to indicate its importance w.r.t student examinations/evaluation, thickness of the node can be used to depict the relative knowledge of the student on the concept. Similarly, edges can be annotated to reveal different kinds of information e.g., thickness of an edge can be used to signify the relatedness between two concepts.

4. EXPERIMENTS AND DISCUSSION

We have implemented the proposed framework in Java and Python and tested it on several examples. Visualization of concept maps is implemented using d3js. In this section, we present the results of one such experiment of generating concept maps using the pdf textbook on databases [2]. Figure 2 depicts a subgraph corresponding to the concepts related to relational algebra. We showed the uncovered concept maps

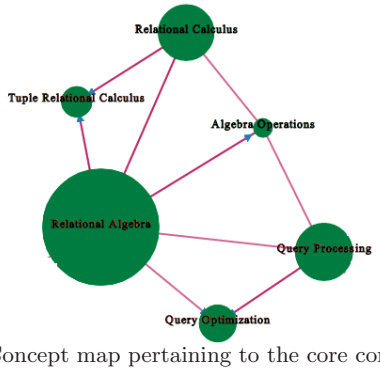


Figure 2: Concept map pertaining to the core concept relational algebra

to a few experts in databases and they mostly agree to the discovered relations. We have applied our approach to several subjects (e.g., operating systems, computer networks etc.) and found that in each of those, we are able to uncover meaningful and important relations. We realize that there is a need for an objective evaluation method to automatically assess the goodness of discovered concept maps, e.g., using gold standard.

5. CONCLUSIONS

Generation of concept maps is an important means of supporting deep understanding of a subject matter. In this paper, we presented an approach for identifying concepts and establishing how closely two concepts are related to each other. We believe that these concept maps enable users to quickly get knowledge about the centrality or importance of each concept and its significance in understanding other concepts. As future work, we would like to further enrich the discovered concept maps with additional information based on the user of the application. For example, upon clicking on a node, teachers/faculty can be provided with information such as the average/distribution score of students on this concept in various tests conducted; students can be provided with links to lecture material, questions/solutions asked in previous exams, etc.

6. REFERENCES

[1] Darina Dicheva and Christo Dichev. Authoring educational topic maps: can we make it easier? In *ICALT*, pages 216–218, 2005.

[2] R. Elmasri and S.B. Navathe. *Fundamentals of database systems*. Pearson Education India, 6 edition, 2010.

[3] Shian-Shyong Tseng, Pei-Chi Sue, Jun-Ming Su, Jui-Feng Weng, and Wen-Nung Tsai. A new approach for constructing the concept map. *Computers & Education*, 49(3):691–707, 2007.

[4] Jorge J. Villalon and Rafael A. Calvo. Concept map mining: A definition and a framework for its evaluation. WI-IAT '08, pages 357–360, 2008.

Integrating Product and Process Data in an Online Automated Writing Evaluation System

Chaitanya Ramineni
Educational Testing Service
Princeton
NJ, 08541
01+609-734-5403
cramineni@ets.org

Tiago Calico
University of Maryland
College Park
MD, 20742
01+301-405-1000
tcalico@umd.edu

Chen Li
Educational Testing Service
Princeton
NJ, 08541
01+609-734-5993
cli@ets.org

ABSTRACT
We explore how data generated by an online formative automated writing evaluation tool can help connect student writing product and processes, and thereby provide evidence for improvement in student writing. Data for 12,337 8th grade students were retrieved from the *Criterion* database and analyzed using statistical methods. The data primarily consisted of automated holistic scores on the student writing samples, and the number of attempts on a writing assignment. The data revealed trends of positive association between the number of revisions and the mean writing scores. User logs were sparse to support study of additional behaviors related to the writing processes of planning and editing, and their relation to the writing scores. Implications for enhancing automated scoring based feedback with learner analytics based information are discussed.

Keywords
Automated scoring, learner analytics, formative writing, automated feedback, process and product

1. INTRODUCTION
The *Criterion*® *Online Writing Evaluation Service* [3], is a web-based writing tool that allows easy collection of writing samples, efficient scoring, and immediate feedback through the *e-rater*® automated essay scoring (AES) engine [2].

Criterion supports essay writing practice with a library of more than 400 essay assignments in multiple discourse modes (expository and persuasive) for students in elementary, middle, and high schools as well as in college. These prompts are used for classroom writing assignments and their scoring is supported by AES models. As a formative writing tool, *Criterion* has several features to facilitate writing processes and help learners improve their writing. These include planning templates, immediate feedback, multiple attempts to revise and edit, and resources such as a Writer's Handbook, a spell checker, a thesaurus and sample essays at different score points. The holistic scoring and feedback in *Criterion* is supported by *e-rater*. The analyses of errors and feedback are available for linguistic features of grammar, usage, mechanics, style and organization and development. There are limited studies on the pedagogical effectiveness of *Criterion* and AES systems in general [1, 5], and examining relation of product and process data for assessing writing quality [4]. Our motivation for this study was to analyze product data (holistic scores) in relation to process data (for revising) to provide evidence for effectiveness of the tool and automated feedback and scoring for improving writing. We report the observed trends for association between the two types of data, the cautions warranted in making strong claims based on these data, and the next steps.

2. METHODS
Data were extracted for 8th grade students for one school year from the *Criterion* database. The data spanned 295 days, and included 12,337 students from 183 schools; a total of 95,261 attempts were made across 41,473 assignments on 2,447 prompts.

Mean holistic scores by the *assignment* and by the *attempt* were examined to relate the revising behavior with improvement in writing scores. The results from the assignment and the attempt level analyses can easily be preliminary indicators of the tool's usefulness and effectiveness, and enhanced data logging capabilities of student actions in the system can provide richer information on writing processes.

3. RESULTS
3.1 Assignment Level
Of the 12,337 students who submitted assignments in the system, a little over 4,000 students submitted only one assignment over the full school year. About half of the students (N=6,663) completed a total of 2 to 6 assignments. A handful of students submitted as many as a total of 15 assignments. We identified groups of students who completed 2 to 5 unique assignments over the period of the full school year (the Ns were small for groups of students completing 6 or more assignments and hence excluded). The assignments in *Criterion* can be scored on a 4-point or a 6-point scale. We analyzed the data for responses evaluated on a 6-point scale only, and hence after filtering out the responses scored on the 4-point scale, the remaining sample size was 5,235 students. It should be noted that within each assignment, a user can have multiple attempts.

Figures 1a and 1b present the trends for the mean writing scores across assignments for the different groups based on the first attempt and the last attempt on the assignment, respectively. We draw quite a few interesting observations from the two graphs. The mean writing scores on the last attempt are always higher than the mean writing scores on the first attempt across all the assignments. Further, the mean writing score on the last attempt of the first assignment (first data point in Figure 1b) is almost always higher than the mean writing score on the first attempt of the fifth assignment (last data point in Figure 1a), suggesting that multiple attempts on an assignment is associated with a higher mean writing score than the total number of assignments completed by a user in the system.

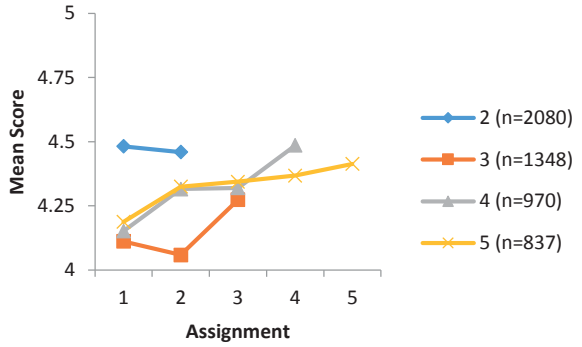


Figure 1a. Mean holistic score on the first attempt, per ordered assignment conditioned on total number of assignments

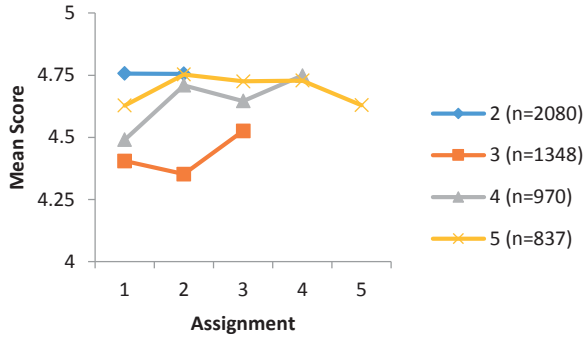


Figure 1b. Mean holistic score on the last attempt, per ordered assignment conditioned on total number of assignments

3.2 Attempt Level

After filtering for responses evaluated using 6-point scale, a total of 34,196 completed attempts were recorded in the system over the full school year. 15,841 of these attempts were instances of one attempt only per assignment. A few students completed as many as 10 attempts on an assignment which is the maximum limit by default. We identified groups of 2 to 6 attempts per assignment that included 16,846 instances (the Ns were small for groups of 7 or more attempts and hence excluded). Figure 2 presents the trends of mean writing scores across attempts for the different groups. The uniform trend of increase in the mean writing scores across the attempts for all the groups once again suggests that the revising process is associated with gains on the writing scores.

4. LIMITATIONS

The data on which trends have been reported were derived from a non-experimental setting. Large groups of students completed only one assignment or submitted only one attempt. Students who did engage in multiple assignments and/or multiple attempts hint at self-selection. The data are unbalanced and highly non-normal, and hence do not support rigorous statistical analyses but rather only lend themselves to exploration for trends.

Server log files were sparse for digital traces of student actions to support nuanced analyses of the corresponding writing processes. Information on students such as background variables is not available in the system. We analyzed data for only one grade level, but it would be of interest to examine if and how the trends based on product data as well as students' usage of the system vary across the different grade levels. Similar analyses of linguistic feature values or error analyses on the product can provide further insight into the process of improvement in student writing.

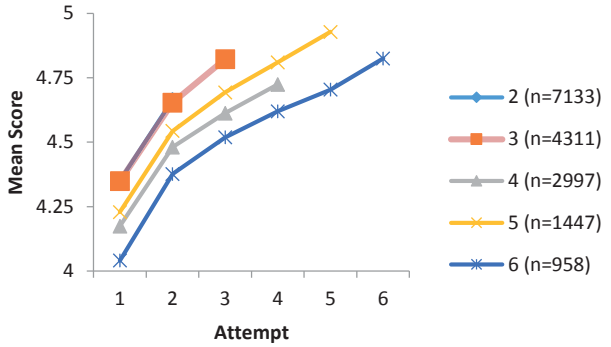


Figure 2. Mean holistic score, per ordered attempt by total number of attempts

5. CONCLUSION

Data currently available from *Criterion* are primarily on the work product; limited data are available for writing processes based on user actions. The additional data from our ongoing work on extension of *Criterion* to capture extended learner usage data will support further analysis of associations between the writing product and the processes, and their relation to change in student writing ability over time. This work has implications for extending application of automated scoring systems in formative contexts with the potential to provide richer feedback on product as well as processes, and enhancing the validity argument for automated scores as supported by response process data.

6. REFERENCES

[1] Attali, Y. 2004. Exploring the feedback and revision features of Criterion. Paper presented at the National Council on Measurement in Education, San Diego, CA.

[2] Attali, Y. & Burstein, J.C. 2006. Automated essay scoring with e-rater v.2. Journal of Technology, Learning, and Assessment, 4(3), (2006), 1–31.

[3] Burstein, J.C., Chodorow, M, & Leacock, C. 2004. Automated essay evaluation: the Criterion online writing service. AI Magazine 25(3), (2004), 27–36.

[4] Deane, P. 2014. Using writing product and process features to assess writing quality and explore how those features relate to other literacy tasks. ETS Research Report No. 14-03. Princeton, NJ: ETS.

[5] Foltz, P., Rosentsein, M., Dronen, N., & Dooley, S. 2014. Automated feedback in a large-scale implementation of a formative writing system: Implications for improving student writing. Paper presented at the annual meeting of the American Educational Research Association, Philadelphia, PA.

Application of Sentiment and Topic Analysis to Teacher Evaluation Policy in the U.S.

Antonio Moretti*
Educator Learning & Effectiveness
Pearson

Kathy McKnight
Educator Learning & Effectiveness
Pearson

Ansaf Salleb-Aouissi
Center for Computational Learning Systems
Columbia University

ABSTRACT

We examine the potential value of Internet text to understand education policy related to teacher evaluation. We discuss the use of sentiment analysis and topic modeling using articles from the New York Times and Time Magazine, to explore media portrayal of these policies. Findings indicate that sentiment analysis and topic modeling are promising methods for analyzing Internet data in ways that can inform policy decision-making, but there are limitations to account for when interpreting patterns over time.

Keywords

Teacher evaluation, topic modeling, sentiment analysis

1. MOTIVATION

In the United States and abroad, teacher evaluation systems are increasingly becoming a common component of school reform efforts. Because teacher effectiveness is central to improving student learning, education policy in the U.S. has targeted teacher evaluation systems, with the rationale that evaluating teachers will lead to improved effectiveness. The result is an often contentious debate among researchers, educators and policy-makers about the utility of these systems in improving teacher effectiveness. Issues include which performance measures to use, how to collect and combine the data, and how it will be used with teachers.

A significant arena for debate about education policy, including teacher evaluations, occurs via the Internet. As the 2013 report "Social Media and Public Policy" notes [Leavy, 2013], use of data produced by Internet users may be useful in understanding policy issues and social problems, and perhaps ultimately, can provide insight to enable governments to develop more informed and better policy. The data may lead to better understanding of policy impact, and could potentially inform the different organizations that deliver public services, such as public education systems.

*Contact author. antonio.moretti@pearson.com

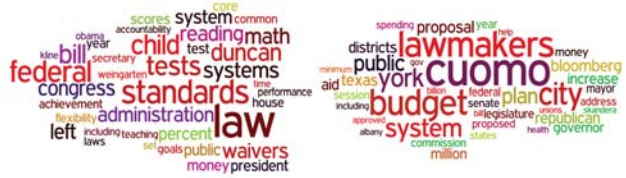
Given the potential value of Internet data to inform policy, our aim for this study is to conduct a preliminary analysis of publicly available Internet data from media outlets reporting on U.S. education policy, to evaluate what might be learned from such data that could inform policy-making regarding teacher evaluation. Therefore, we narrowed the focus to two popular media sources that cover national as well as local education policy – the NY Times, and Time Magazine–to analyze public sentiment and topics of concern regarding education policy focused on teacher evaluation. Given the increased emphasis on teacher evaluations over the past decade, we gathered data from 2004 - 2014. We used two approaches for analyzing data from the online media articles: a topic modeling approach [Blei, 2012] and sentiment analysis [Liu, 2010, pan,]. The research questions we addressed included:

1. What trends, if any, exist in public sentiment regarding teacher evaluation policy over the past decade?
2. What are the recurring topics most associated with media portrayal of teacher evaluation policies?

2. DATA COLLECTION AND ANALYSIS

We used the NY Times API and Time Magazine search query using "teacher evaluation" as the search term. Because there are no tools for collecting the full NY Times and Time Magazine articles,we scraped the websites after retrieving the relevant URLs. We retrieved a total of 348 articles on "teacher evaluation" from the NY Times during the period 2004 to 2014, and 292 articles from Time Magazine during the same period. We examined the articles for their relevance and removed those for which the focus was not primarily on teacher evaluation. The resulting dataset included 171 NY Times articles from 2009 to 2014, and 45 Time Magazine Articles from 2010 to 2014.

For the current study, we used the "topicmodels" package in R [Grün and Hornik, 2011]. We compared two variants of topic modeling: latent dirichlet allocation (LDA) and Correlated Topic Models (CTM). Both approaches are based on Blei [Blei et al., 2003, Blei and Lafferty, 2007]. To determine the number of topics to specify, we used the perplexity score.For our analyses, we specified a ten topic model, i.e. we set $k = 10$ to interpret results. In addition to the entropy measure, we used word clouds to display and make



(a) Topic 1 (b) Topic 2

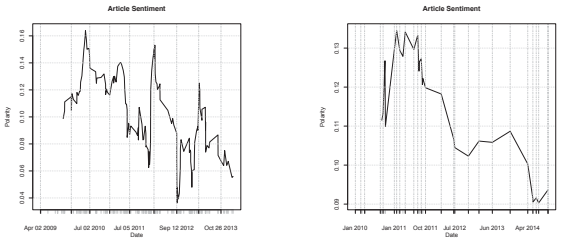
Figure 1: Word clouds for generated from NY Times articles.

sense of the topics generated from topic models. Figure illustrates word clouds for topic 1 and 2 generated from the New York Times articles. The topics that appeared to dominate the NY Times reporting included focus on federal requirements for teacher evaluation systems (e.g., reliance on student test data and relatedly, value-added models, for evaluating teachers); the impact of those requirements on teachers at both a federal and local (NYC) level, e.g., accountability, merit pay, lay offs and budgets; and the reaction of teacher unions to federal and local legislation (e.g., Chicago's teachers strike). In Times Magazine, where coverage of teacher evaluation policy was often combined with coverage of other federal education policies, the focus appeared to be on student achievement testing; changing education policies by the Obama Administration and in Washington DC, led by DC's former Chancellor of Education Michelle Rhee; and policy proposals during the 2012 presidential campaign. Teacher union reactions to teacher evaluation policy were also of focus, including the Chicago teachers strike.

We use the Natural Language Toolkit (NLTK) [Bird, 2006], a leading python platform to harvest textual data. The sentiment analysis tool in NLTK uses naive Bayes classifiers trained on both twitter sentiment as well as movie reviews. In Figure 2, a time series of the sentiment polarity of both the New York Times (left) and Times Magazine (right) articles is presented for 2009 - 2014. We used a simple moving average to plot the sentiment over time. In these graphs, we observe a similar trend in both the NY Times and Times Magazine articles. In both, we see somewhat similar peaks and troughs, as well as a similar trend of decreasing positive sentiment from 2010 to 2014.

3. DISCUSSION & FUTURE WORK

A number of federal and local (to NY) events took place over that period of time, that could be related to the sentiment trends. Nationally, the Obama Administration's Race To the Top (RTTT) legislation was initiated in July 2009, which among other policies, required states to develop and implement teacher evaluation systems that included student achievement as a "significant" component of a teacher's effectiveness rating. In 2010, RTTT was rolled out and the states awarded funding were announced. The state of New York was awarded 700M dollars in August, 2010. A result of this legislation was a contentious battle between lawmakers and the teachers union over the details of the evaluation system, among other policies. In September 2012 in Chicago, teachers took to the streets and went on strike against a range of education policies, including the teacher evaluation system that was to be put in place. NYC and the teachers union settled on an evaluation system in March, 2013.



(a) Topic 1 (b) Topic 2

Figure 2: Article Sentiment over Time

In the case study for this paper, issues regarding the use of student test scores for evaluating teachers; the response of teacher unions to federal and local teacher evaluation system requirements; and the budgets for implementing these systems were just some of the more prominent issues reflected in the results. Our ultimate goal is to advance the understanding of the impact of new policies on the well-being of public schools and teachers. While the methodology is promising, it needs to be harnessed through a useful visualization interface to facilitate the exploration and analysis of the topics produced to make it more useful to leverage in decision making. We acknowledge that there are limitations and potential problems with these approaches. A known challenge is choosing the granularity level of the topics that is related to the number of topics k provided as a parameter. A second challenge is in the interpretation and labeling of the derived topics that require a manual human intervention. In some cases, what is rated as positive or negative analytically might not reflect how human raters would code those words. Moreover, in our example, although sentiment appeared to decline in the negative direction, it still remained on the positive end of the polarity continuum.

4. REFERENCES

[pan,]

[Bird, 2006] Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

[Blei and Lafferty, 2007] Blei, D. M. and Lafferty, J. D. (2007). A correlated topic model of science. *AAS*, 1(1):17–35.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Grün and Hornik, 2011] Grün, B. and Hornik, K. (2011). topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30.

[Leavy, 2013] Leavy, J. (2013). Social media and public policy: What is the evidence? Technical report, Alliance for Useful Evidence.

[Liu, 2010] Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*.

Defining Mastery: Knowledge Tracing Versus N- Consecutive Correct Responses

Kim Kelly
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
508-461-6386
kkelly@wpi.edu

Yan Wang &
Tamisha Thompson
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
Ywang14@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
508-831-5569
nth@wpi.edu

ABSTRACT
Knowledge tracing (KT) is well known for its ability to predict student knowledge. However, some intelligent tutoring systems use a threshold of consecutive correct responses (N-CCR) to determine student mastery, and therefore individualize the amount of practice provided to students. The present work uses a data set provided by ASSISTments, an intelligent tutoring system, to determine the accuracy of these methods in detecting mastery. Study I explores mastery as measured by next problem correctness. While KT appears to provide a more stringent threshold for detecting mastery, N-CCR is more accurate. An incremental efficiency analysis reveals that a threshold of 3 consecutive correct responses provides adequate practice, especially for students who reach the threshold without making an error. Study II uses a randomized- controlled trial to explore the efficacy of various N-CCR thresholds to detect mastery, as defined by performance on a transfer question. Results indicate that higher thresholds of N-CCR lead to more accurate predictions of performance on a transfer question than lower thresholds of N-CCR or KT.

Keywords
Intelligent Tutoring System, Knowledge Tracing, Mastery Learning.

1. INTRODUCTION

Intelligent tutoring systems are known for their ability to personalize the learning experience for students. One way that learning is individualized is by providing just the right amount of practice to meet the student's needs. Determining the correct amount of practice is critical because over-practice might bore students and take an un-necessarily long time, while under-practice might not provide enough opportunities for a student to learn a skill. To determine the correct amount of practice, systems must identify the point in time when students have learned the skill, otherwise referred to as reaching mastery.

Defining mastery may vary between systems. One measure of mastery includes next problem correctness, another is performance on a transfer question, and yet another is performance on a delayed retention test. Some systems rely on knowledge tracing (KT) [1-2], others use a predetermined number of consecutive correct responses (N-CCR) [3, 4, 9}. In each case, mastery status is used by the system to determine the end of an assignment.

2. METHODOLOGY
This research is comprised of two studies, the first was a data analysis of large data sets provided by ASSISTments, and the second was a randomized controlled trial. Study I of the present study leverages data generated by an intelligent tutoring system to explore the ability of N-CCR and KT to detect mastery. Mastery will be measured by next problem correctness. Additionally, an incremental efficiency analysis will also be presented that sheds light on the number of additional questions students must answer to reach a given threshold.

Next problem correctness is arguably a weak measure of mastery as slips are possible. A measure of more robust learning is performance on a transfer task [10]. Therefore, in Study II, a randomized-controlled trial was conducted to compare the accuracy of different potential thresholds of number of consecutive correct responses. This data was then used to further explore KT predictions, compared to N-CCR in an attempt to determine which method should be used in intelligent tutoring systems who rely on mastery to determine amount of practice.

3. RESULTS

3.1 NCCR
When mastery is defined by next problem correctness, results indicate that 3-CCR is an adequate threshold for accurately detecting mastery. Table 1 shows that 80% of students who answer three questions correctly, go on to answer the fourth and fifth correctly as well.

Table 1: Percentage of students with each response combination of the fourth and fifth question following 3-CCR.

3 Consecutive No Errors		Fourth Question	
		Incorrect	Correct
Fifth Question	Incorrect	1.8% (5)	9.8% (24)
	Correct	8.4% (28)	80.0% (228)

When mastery is defined by performance on a transfer question, results indicate that 5-CCR (Table 3) more accurately detects mastery than 3-CCR (Table 2). Accuracy is defined by the percentage of students who met the threshold and were successful on the transfer questions combined with the percentage of students who failed to meet the threshold and answered the transfer questions incorrectly. Identifying students who met the threshold yet answered the transfer incorrectly are considered false positives and students who answered the transfer question

correctly yet failed to meet the threshold are considered false negatives.

Table 2: Student performance on transfer question based on 3-CCR.

Percent(Number) of students	Threshold Met	Threshold Not Met
Transfer Correct	46%(17)	0%
Transfer Incorrect	43%(16)	11%(4)

Table 3: Student performance on transfer question based on 5-CCR.

Percent(Number) of students	Threshold Met	Threshold Not Met
Transfer Correct	43%(16)	8%(3)
Transfer Incorrect	19%(7)	30%(11)

3.2 KT

When mastery is defined by next problem correctness, results indicate that KT is comparable to 3-CCR in accurately detecting mastery for students who do not make an error (Table 4).

Table 4: Accuracy of KT detecting mastery for students who answered three consecutive questions correctly without an error. (n=287)

	Threshold Met (>95%)	Threshold Not Met (<95%)
Next Question Correct	80.5% (231)	9.4% (27)
Next Question Incorrect	8.4% (24)	1.7% (5)

When mastery is defined by performance on a transfer question, results indicate that KT is comparable to 3-CCR, but less accurate than 5-CCR (Table 5).

Table 5: . Student performance on the transfer question based on KT's 95% threshold.

Percent(Number) of students*	Threshold Met	Threshold Not Met
Transfer Correct	42%(31)	7%(5)
Transfer Incorrect	39%(29)	12%(9)

3.3 Incremental Efficiency Analysis

Using the data generated from the students reaching the 5-CCR threshold, we determined how many additional questions were required to reach each incremental threshold. This provides insight into the tradeoff between potential increased mastery detection and time consumption, as measured by number of questions completed. 3-CCR is a sufficient threshold, as over 90% students go on to reach the higher threshold. Of the students who reached the final 5-CCR threshold, 90% of them reached it without an error. Those who made at least one error, tended to reach the threshold with N attempts following the error. This suggests that the error was a slip.

4. DISCUSSION

Accurately predicting or detecting mastery status is critical to intelligent tutoring systems, because the amount of practice provided to students depends on this. An overly cautious prediction will lead to unnecessary practice (false negatives), while less strict criteria will not provide enough (false positives).
N-CCR, specifically 3-CCR, is a simple, yet effective way to determine mastery within an ITS. This threshold has been found to predict next problem correctness with at least 80% accuracy. However, when predicting performance on a transfer task, a higher threshold (5-CCR) is more effective. Both thresholds of N-CCR were more accurate than the more complicated method, knowledge tracing, when determining mastery.

5. ACKNOWLEDGMENTS

We thank multiple NSF grant (ACI-1440753, DRL-1252297,DRL-1109483,DRL-1316736 & DRL-1031398) , the US Dept of Ed's (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation.

6. REFERENCES

[1] Corbett, A., Anderson, J. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction, 4*, 253-278.
[2] Fanscali, Stephen E., Nixon, Tristan, & Ritter, Stephen (2013). Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. *Proceedings of the 6th International Conference on Educational Data Mining.* D'Mello, S., Calvo, R., Olney, A. (Eds). 35-42.
[3] Faus, M. (2014). Improving Khan Academy's student knowledge model for better predictions. *MattFaus.com* [web log]. Retrieved October, 2014, from http://mattfaus.com/2014/05/improving-khan-academys-student-knowledge-model-for-better-predictions/
[4] Feng, M., Heffernan, N. T., Koedinger, K. R.: Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19, 243-266 (2009)
[9] Hu, D. (2011). How Khan Academy is using Machine Learning to Assess Student Mastery. *David-Hu.com* [web log]. Retrieved October, 2014, from http://david-hu.com/2011/11/02/how-khan-academy-is-using-machine-learning-to-assess-student-mastery.html

A Toolbox for Adaptive Sequence Dissimilarity Measures for Intelligent Tutoring Systems

Benjamin Paassen
CITEC Center of Excellence
Bielefeld, Germany
bpaassen@techfak.uni-bielefeld.de

Bassam Mokbel
CITEC Center of Excellence
Bielefeld, Germany
bmokbel@techfak.uni-bielefeld.de

Barbara Hammer
CITEC Center of Excellence
Bielefeld, Germany
bhammer@techfak.uni-bielefeld.de

ABSTRACT

We present the *TCS Alignment Toolbox*, which offers a flexible framework to calculate and visualize (dis)similarities between sequences in the context of educational data mining and intelligent tutoring systems. The toolbox offers a variety of alignment algorithms, allows for complex input sequences comprised of multi-dimensional elements, and is adjustable via rich parameterization options, including mechanisms for an automatic adaptation based on given data. Our demo shows an example in which the alignment measure is adapted to distinguish students' Java programs w.r.t. different solution strategies, via a machine learning technique.

1. INTRODUCTION

Systems for computer-aided education and *educational data mining* (EDM) often process complex structured information, such as learner solutions or student behavior patterns for a given learning task. In order to abstract from raw input information, the given data is frequently represented in form of sequences, such as (multi-dimensional) symbolic strings, or sequences of numeric vectors. These sequences may represent single solutions, as in some *intelligent tutoring systems* (ITSs) [2, 6]; or may encode time-dependent data, like learner development or activity paths [1, 7].

Once a meaningful sequence representation is established, there are many possibilities to process sequential data with existing machine learning or data mining tools. A crucial component for this purpose is a (dis)similarity measure for pairs of sequences, which enables operations like finding closest matches in a given data set, clustering all instances, or visualizing their neighborhood structure [5]. One particularly flexible approach to determine the (dis)similarity of sequences is *sequence alignment* [3].

For applications in the context of EDM and ITSs, sequence alignment offers two key features: On the one hand, the structural characteristics of sequences are taken into account, while calculation remains efficient, even with complex parameterization options. On the other hand, alignment provides an intuitive matching scheme for a given sequence pair, since both sequences are extended, so that similar parts are *aligned*. However, we believe the full potential of sequence alignment is rarely utilized in EDM or ITSs.

Acknowledgments: Funding by the DFG under grant numbers HA 2719/6-1 and HA 2719/6-2 and the CITEC center of excellence is gratefully acknowledged.

2. ALIGNMENT TOOLBOX

We present the *TCS Alignment Toolbox*[1], an open-source, Matlab-compatible Java library, which provides a flexible framework for sequence alignments, as follows:

Multi-dimensional input sequences are possible, such that every element of the sequence can contain multiple values of different types (namely discrete symbols, vectors or strings).

A **variety of alignment variants** is implemented, covering common cases, such as *edit distance*, *dynamic time warping* and *affine sequence alignment* [3].

The **parameterization** of the alignment measure is defined by costs of operations (replacement, insertion, and deletion) between sequence elements, which can be adjusted by the user, or left at reasonable defaults. Users can even plug in custom functions to yield meaningful problem-specific costs.

A **visualization feature** displays the aligned sequences in a comprehensive HTML view, as well as the dissimilarity matrix for an entire set of input sequences.

An approximate **differential of the alignment functions** w.r.t. its parameters is provided, which enables users to automatically tune the rich parameter set with gradient-based machine learning methods, e.g. to facilitate a classification [4].

In this demo, we present an example for a set of real student solutions for a Java programming task: After programs are transformed to sequences, the parameters of an alignment algorithm are automatically adapted to distinguish between different underlying solution strategies, and the resulting alignments are visualized. Thus, the adapted measure improves the classification accuracy for the given data.

3. REFERENCES

[1] S. Bryfczynski, R. P. Pargas, M. M. Cooper, M. Klymkowsky, and B. C. Dean. Teaching data structures with besocratic. In *ITiCSE 2013*, pages 105–110. ACM, 2013.

[2] S. Gross, B. Mokbel, B. Hammer, and N. Pinkwart. How to select an example? A comparison of selection strategies in example-based learning. In *ITS 2014*, pages 340–347, 2014.

[3] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, New York, NY, USA, 1997.

[4] B. Mokbel, B. Paassen, F.-M. Schleif, and B. Hammer. Metric learning for sequences in relational LVQ. *Neurocomputing*, 2015. (accepted/in press).

[5] E. Pekalska and B. Duin. *The Dissimilarity Representation for Pattern Recognition*. World Scientific, 2005.

[6] E. R. Sykes and F. Franek. A prototype for an intelligent tutoring system for students learning to program in Java (TM). *IASTED 2003*, pages 78–83, 2003.

[7] N. van Labeke, G. D. Magoulas, and A. Poulovassilis. Searching for "people like me" in a lifelong learning system. In *EC-TEL 2009*, volume 5794 of *LNCS*, pages 106–111. Springer, 2009.

[1] Available at `http://opensource.cit-ec.de/projects/tcs`

Carnegie Learning's Adaptive Learning Products

Steven Ritter
Ryan Carlson
Michael Sandbothe
Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, 20th Floor
Pittsburgh, PA 15219 USA
1.888.851.7094 {x122, x219}
{sritter, rcarlson, msandbothe,
sfancsali}@
carnegielearning.com

ABSTRACT

Carnegie Learning, developers of the widely deployed Cognitive Tutor, has been working on several new adaptive learning products. In addition to demoing the Cognitive Tutor, an educationally effective intelligent tutoring system for mathematics that has been the subject of a great deal of educational and educational data mining research, we demo two iPad apps, an equation solving app that recognizes hand writing and a game for developing math fluency using fraction comparison tasks. A wide variety of datasets over the years have been analyzed from the Cognitive Tutor, and in recent years several new features have been introduced that may be important to researchers. This demonstration will introduce those unfamiliar with Cognitive Tutor to the system and serve as a refresher for those unaware of recent developments. It will also introduce our new iPad apps to researchers.

Keywords

Cognitive Tutor, intelligent tutoring systems, real-world implementation, mathematics education, educational games, iPad, mathematics fluency, fractions, decimals, multiple representations, equation solving, cognitive modeling

1. COGNITIVE TUTOR

Carnegie Learning's Cognitive Tutor (CT) [7] is one of the most widely used intelligent tutoring systems (ITSs) in the world, with hundreds of thousands of users in middle schools, high schools, and universities throughout the United States and abroad. CT has been demonstrated effective in one of the largest randomized trials of its kind involving educational software, providing substantive and significant improvement in learning gains, compared to a control group using traditional textbooks, in the second year of implementation for a large cohort of high school students from diverse regions of the United States [6].

A variety of datasets providing information about learner interactions with the CT have been made available by Carnegie Learning via the Pittsburgh Science of Learning Center LearnLab's DataShop repository [5]; the learning sciences community and others have used these and other datasets in a correspondingly wide variety of educational and educational data mining (EDM) research projects, including many throughout the history of the *International Conference on EDM*. Some datasets used are from relatively older versions of the CT software. Even relatively old data can enable discovery and insight into issues likes improving cognitive models and improving the predictive accuracy of models of student behavior, but as can be expected, CT, like any other piece of widely deployed software, evolves over time. Elements of this evolution may impact the types of substantive conclusions that can be drawn from CT data or contribute to creative new modeling approaches and target educational phenomena. In this demonstration, we will provide an overview of the basic interface of the CT and its approach to mathematics education as well as highlighting several newer features that have been deployed in the last few years. We will also, as appropriate, highlight several nuances and issues that arise when CT and Carnegie Learning's middle school math product based on CT, called MATHia, are deployed in real-world classrooms. Some of these nuances and issues may have important implications for how EDM analyses are conducted using CT data.

Our demo will provide a general overview with CT and focus on the following features of CT and MATHia: lesson content and manipulatives, step-by-step examples, review mode, promotion & placement changes, interest area & name customization (MATHia), and math "Fluency Challenge" Games (MATHia).

2. AN IPAD RACING GAME TO ENHANCE MATH FLUENCY

Developers at Carnegie Learning are also developing an iPad car racing game (Figure 1) to enhance math fluency for tasks like comparing fractions. The game integrates with the Hyper-Personalized Intelligent Tutoring (HPIT) system [4], a distributed web service plugin architecture that enables "on-the-fly" personalization based on (non-)cognitive factors. Gameplay is predicated on learners rotating the iPad to direct a car to the right, left, and in between "flags" that display values of fractions (or decimals, etc.) based on whether a value displayed on the car is greater than or less than values displayed on flags, creating a sort of number line on the game's "road."

Time pressure, introduced via a countdown clock, serves gameplay and cognitive functions. Time pressure on tasks like fraction comparison will encourage learners to develop dynamic

strategies to carry out such tasks (e.g., imagining slices of a pie vs. finding common denominators). Learners' successful adoption of diverse strategies is a marker of math fluency that will decrease working memory load on such tasks. We posit that fluent math learners are more likely to succeed in more advanced math.

Game content and behavior are configurable to allow education researchers, without programming, to rapidly prototype and build a range of experiments. Researchers can, for example, specify number sequences encountered as well as "level" structure that groups similar content together. We support in-game feedback (e.g., text displayed after questions, pausing after incorrect actions for review) via an XML run-time scripting engine.

Figure 1. Sample problem: The player's value is 1/9, and since 1/9 < 1/7 the player moves to the left lane before passing the flags.

A conceivable experiment uses multiple graphical representations to develop fluency [1]. Curricula can begin with a level containing common numerator fractions, then common denominator fractions, and then mixed fractions. Scripting provides for dynamic annotations of each fraction with pie slice or number line images above flags to help players visualize the comparison (e.g., loading web images and reacting to each level's content). Help can be offered only when a student is struggling (e.g., making at least one error), and HPIT can drive A/B tests, distributing content/scripts to control and experimental groups.

3. AN IPAD APP FOR EQUATION SOLVING

Researchers at Carnegie Learning are also working on an iPad app to support math equation solving practice. The app combines technology from CT with an interface that recognizes human handwriting (Figure 2). Following the lead of CT and building on earlier work on handwriting-based tutors [2], the app provides context sensitive feedback and hints while also providing the capability to "trace" student knowledge using, like CT, the Bayesian Knowledge Tracing (BKT) [3]. Integrating the app with HPIT provides the ability to adapt to cognitive factors (e.g., BKT) and non-cognitive factors (e.g., grit, self-efficacy, etc.).

The app will advance student learning about equation solving and our understanding of that learning in at least two ways. First, handwriting recognition will provide for an experience that is more akin to a traditional "pencil and paper" approach to equation solving practice than the approach provided by CT in which actions like "combining like terms" to manipulate sides of an equation are chosen from a drop-down menu. Second, logging such equation solving will provide rich data to better understand the learning of equation solving in this more natural setting.

Moving away from the menu-based CT approach introduces challenges. Handwritten equation solving allows for a variety of math errors that simply are not allowed by CT. Further, new knowledge components (or skills) must be introduced to the cognitive/skill model for this app; skills, for example, related to the understanding of equality (e.g., that the equation symbol must persist from line to line as the student works toward an equation solution) should be tracked. Such skills are not tracked in CT's menu-based equation solving because the equation symbol persists from step-to-step in CT. Comparing skill models and learner performance across platforms is a key area for future research; translation of skill models across platforms is an important issue as technology permeates teaching and instruction.

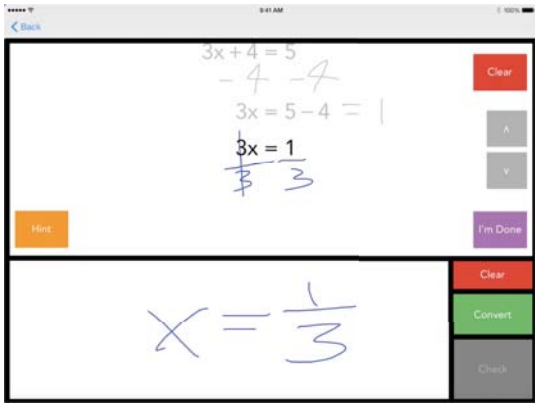


Figure 2. A user solves the equation 3x+4 = 5, writing the final step of the equation as x = 1/3.

4. ACKNOWLEDGMENTS

App development is funded by the U.S. Department of Defense Advanced Distributed Learning Initiative Contract #W911QY-13-C-0026.

5. REFERENCES

[1] Ainsworth, S. 2006. DeFT: A conceptual framework for considering learning with multiple representations. *Learn. Instr.* 16 (Jun. 2006), 183-198.

[2] Anthony, L., Yang, J., Koedinger, K.R. 2014. A paradigm for handwriting-based intelligent tutors. *Int. J. Human-Computer Studies*, 70, 866–887.

[3] Corbett, A.T., Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap.* 4, 253-278.

[4] Fancsali, S.E., Ritter, S., Stamper, J., Nixon, T. 2013. Toward "hyper-personalized" Cognitive Tutors: Non-cognitive personalization in the Generalized Intelligent Framework for Tutoring. In *AIED 2013 Workshops Proceedings Volume 7* (Memphis, TN, July, 2013). Sun SITE Central Europe (CEUR), 71-79.

[5] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2011. A data repository for the EDM community: the PSLC DataShop. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, & R.S.J.d. Baker, Eds. CRC, Boca Raton, FL.

[6] Pane, J., Griffin, B. A., McCaffrey, D. F., Karam, R. 2014. Effectiveness of Cognitive Tutor Algebra I at scale. *Educ. Eval. Policy. An.* 36 (2014), 127-144.

[7] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14 (2007), 249-255.

SAP: Student Attrition Predictor

Devendra Singh Chaplot
Samsung Electronics Co., Ltd.
Seoul, South Korea
dev.chaplot@samsung.com

Eunhee Rhim
Samsung Electronics Co., Ltd.
Seoul, South Korea
eunhee.rhim@samsung.com

Jihie Kim
Samsung Electronics Co., Ltd.
Seoul, South Korea
jihie.kim@samsung.com

ABSTRACT

Increasing rates of student drop-outs with increase in popularity of Massive Open Online Courses (MOOCs) makes predicting student attrition an important problem to solve. Recently, we developed an algorithm based on artificial neural network for predicting student attrition in MOOCs using student sentiments. In this paper, we present a web-based tool based on our algorithm which can be used by educators to predict and reduce attrition during a course and by researchers to design and train their own system to predict student attrition.

Keywords

Student Attrition, MOOC, Sentiment Analysis, Neural Network, Educational Data Mining, Student Drop-out

1. OVERVIEW

Growing popularity of MOOCs is attributed to their accessibility, scalability and flexibility. With scalability, MOOCs also provide huge amounts of data of student activity which can be used to predict their behavior. We have developed an algorithm to predict student attrition [4] which uses clickstream log and forum posts from MOOCs to extract features such as number of page views, clicks, study sessions, etc. as suggested by previous studies [1, 3, 5, 6]. A unique feature used by our algorithm is student sentiments in forum posts, which is calculated using lexicon-based Sentiment Analysis with SentiWordNet 3.0 [2] as the knowledge resource. The values of all these features for current week are passed as inputs into an artificial neural network, whose output indicates whether student is going to drop out in the following week. Using data from Coursera course 'Introduction to Psychology', we get 74.4% accuracy with false negative ratio of 0.136, leading to a Cohen's Kappa value of 0.435.

2. STUDENT ATTRITION PREDICTOR

We present a web tool having three interfaces for educators and researchers to predict and study student attrition.

2.1 Sentiment Analysis

Sentiment Analysis of student's forum posts is the unique feature which wasn't used by previous algorithms and improves the Cohen's Kappa value of our algorithm by about 13%. Effectiveness of using sentiment analysis can be seen by the changes in results from neural network when student sentiments are added as input. Our tool also provides option to get the Sentiment score of any student's forum post.

2.2 Pre-trained Neural Network

Users have the option to use our pre-trained neural network to predict student drop-out. This allows our tool to be used freely by educators to predict student attrition. Since we predict whether student is going to drop-out in the following week and not whether student is going to complete the course, our algorithm pin-points the exact week when student is predicted to drop-out and thus, educators can use our tool during the course in order to take necessary student-specific actions to prevent or reduce attrition. Apart from MOOCs, Student Attrition Predictor can also be used by traditional classroom setting educators, using digital mediums for study and interaction in schools, which are becoming increasingly popular in recent years.

2.3 Design new Neural Network

Our tool also provides an interactive graphical interface for the users to design their own unique neural network. A screenshot of design interface is shown in Figure 1. It shows an input panel, training and testing data panels, a neural network design canvas and a results panel. The process of using Design interface can be divided into 3 phases:-

- **Design:** Users can add their own nodes in the 'Input' panel and select any number of hidden layer nodes. The canvas in the middle of Figure 1 shows the structure of designed neural network.
- **Train:** Training data can be uploaded in 'Training Data' panel and used to train the designed neural network. Options for selecting number of training iterations, classification boundary and learning heuristic (like back-propagation, resilient propagation, etc.) for training Neural Network will also be provided.
- **Test:** After training, individual input values can be entered in the input panel or test data can be uploaded in 'Test Data' panel to get results from trained neural network. 'Results' panel shows metrics such as Accuracy, False Negative Rate and Cohen's Kappa value.

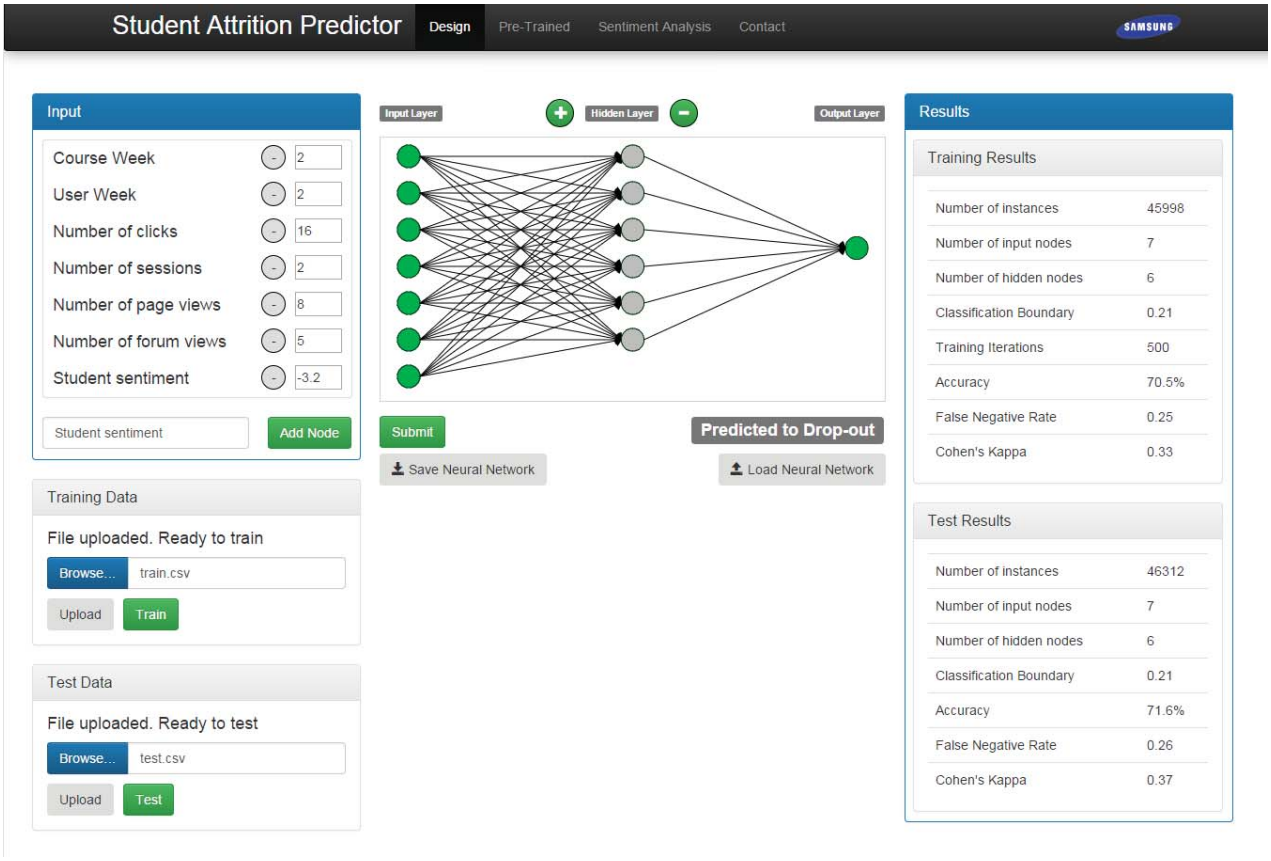


Figure 1: Screenshot of Design Interface of Student Attrition Predictor

This interface is especially useful for researchers who can decide the input features and structure of their own neural network, train and test it by uploading their own data and optimize the parameters and learning heuristic according to their application. The designed and trained neural network can be saved and loaded into the tool at any point.

3. CONCLUSION

There has been lot of research in recent years on predicting student attrition. In contrast to many studies trying to find reasons behind attrition, we focus on predicting and reducing attrition. Student Attrition Predictor not only predicts student drop-out, but also identifies the precise week when student is likely to drop-out in order to reduce attrition during the course. To the best of our knowledge, there is no direct way for educators to benefit from years of research on predicting student attrition. This tool acts as a medium for educators to directly utilize our research in this field. The tool also provides an easy graphical interface to researchers for further experiments.

4. REFERENCES

[1] B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong. Predicting Attrition Along the Way: The UIUC Model. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 55–59, Doha, Qatar, October 2014.

[2] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.

[3] G. Balakrishnan. Predicting student retention in massive open online courses using hidden markov models. Master's thesis, EECS Department, University of California, Berkeley, May 2013.

[4] D. S. Chaplot, E. Rhim, and J. Kim. Predicting student attrition in moocs using sentiment analysis and neural networks. In *Proceedings of AIED 2015 Fourth Workshop on Intelligent Support for Learning in Groups*, Madrid, Spain, 2015.

[5] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting MOOC Dropout over Weeks Using Machine Learning Methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, Doha, Qatar, October 2014.

[6] M. Sharkey and R. Sanders. A Process for Predicting MOOC Attrition. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 50–54, Doha, Qatar, October 2014.

DOCTORAL CONSORTIUM PAPERS

Dynamic User Modeling within a Game-Based ITS

Erica L. Snow
Arizona State University
Tempe, AZ, 85283
Erica.L.Snow@asu.edu

ABSTRACT

Intelligent tutoring systems are adaptive learning environments designed to support individualized instruction. The adaptation embedded within these systems is often guided by user models that represent one or more aspects of students' domain knowledge, actions, or performance. The proposed project focuses on the development and testing of user models within the iSTART-2 intelligent tutoring system, which will be informed by dynamic methodologies and data mining techniques. My previous work has used post hoc dynamic methodologies to quantify optimal and in-optimal learning behaviors within the game-based system, iSTART-2. I plan to build upon this work by conducting dynamical analyses in real-time to inform the user models embedded within iSTART-2. I am seeking advice and feedback on the statistical methods and feature selection that should included within the new dynamic user model. The implications of this approach for both iSTART-2 and the EDM field are discussed.

Keywords

Intelligent Tutoring Systems, Dynamic Analysis, Adaptation, Log Data, User Models

1. INTRODUCTION

Intelligent tutoring systems (ITSs) are adaptive learning environments that provide customized instruction based on students' individual needs and abilities [1]. ITSs are typically more advanced than traditional computer-assisted training in that they *adapt* to the users' performance and skill levels [2]. The customizable nature of ITSs has resulted in the successful integration of these systems into a variety of settings [3-5].

One hypothesized explanation for the widespread success of these systems is that ITSs provide individualized feedback and adjust content based on the unique characteristics of each student or user. This pedagogical customization allows students to progress through learning tasks at a pace that is appropriate to their individual learning model [6]. It also ensures that students are not only learning at a shallow procedural level, but they are gaining deeper knowledge at an appropriate pace.

One way in which ITSs store and represent information about learners is via *user models*. User models embedded within ITSs incorporate detailed representations of learners' knowledge, affect, and cognitive processes [7]. It is important to note that these models are often continuously updating throughout the students' interaction within the system. Thus, potentially, every student action or decision made within the system contributes to more accurate and holistic user models. Although this concept seems to be intuitive, researchers often struggle to determine what information belongs within the models and how to optimally quantify the dynamic nature of that information.

In prior work, my colleagues and I have proposed that dynamical systems theory and associated analysis techniques are useful tools for examining behavioral patterns and variations within ITSs [8,9]. Indeed, dynamic systems theory affords researchers a unique means of quantifying patterns that emerge from students' interactions and learning behaviors within an ITS. This approach treats time as a critical variable by focusing on the complex and fluid interactions that occur within a given environment rather than treating behavior as static (i.e., set or unchanging), as is customary in many statistical approaches. In the proposed work, I hypothesize that dynamical methodologies have strong potential to inform user models by quantifying changes in students' interactions and learning behaviors across time. This quantification and modeling of behavior can inform decisions about how content and feedback should be presented to each student based on their current learning trajectory. The overall goal of the proposed work is to test the utility of real-time dynamic analyses as a way to inform user models about optimal (and non-optimal) learning behaviors within a game-based ITS.

1.1 iSTART-2

Interactive Strategy Training for Active Reading and Thinking-2 (iSTART-2) is a game-based ITS designed to improve high school students' reading comprehension via self-explanation strategies [10]. In previous studies, iSTART-2, and its predecessors, have been shown to be effective at improving students' self-explanation quality and reading comprehension ability [11, 12].

iSTART-2 consists of two phases: self-explanation training and game-based practice. During training, students watch a series of videos that introduce them to and provide examples of self-explanations strategies. After students view these videos, they transition to practice (see Figure 1 for a screenshot of the game-based practice interface). During practice, students are able to interact with a suite of mini-games, personalizable features, and achievement screens [13]. The game-based practice embedded within iSTART-2 is designed to promote the generation and identification of self-explanation strategies. Within these practice games students are exposed to game mechanics that serve as a form of feedback on their understanding of the self-explanation strategies (see [11] for more details).

The interface of iSTART-2 uniquely affords students substantial agency and control over their learning path by allowing them to choose how they engage with the practice environment [9]. Such freedom also affords researchers with the opportunity to explore and *model* how and when students engage with these features and activities, and to explore the implications of such choices (i.e., how they affect performance and learning).

Figure 1. Screen shot of iSTART-2 Selection Menu

1.2 Current Work

My doctoral research will use educational data mining methods to inform and build dynamic student models within game-based ITSs such as iSTART-2 [13]. Specifically, this study will explore how dynamic techniques such as Hurst exponents and Entropy analysis can be used in real-time to quantify students' behaviors, performance, and cognition while they learn within iSTART-2. Analyses of students' logged choices have been shown to be a *blueprint* regarding successful and unsuccessful behaviors for learning [13, 15]. Therefore, the logged information from iSTART-2 will be used in conjunction with dynamical analysis techniques as a means to quantify various types of in-system behaviors and their impact on learning outcomes in real-time. This information will then be used to adapt the pedagogical content students are exposed to.

2. Proposed Contributions of Current Work

The current work has both *local* and *global* implications. Locally, the development of dynamic user models will improve iSTART-2 pedagogy. Currently, iSTART-2 has limited user models (only guides self-explanation feedback) embedded within the system. Thus, the inclusion of a dynamic user model is expected to improve system feedback and guide the content presentation provided to students. For instance, one research question that arises from this work is how to support optimal learning trajectories for every student. Dynamic user models have the potential to recognize non-optimal learning behaviors and provide feedback or navigate students toward more effective learning behaviors within the practice environment. Thus, it is hypothesized that the implementation of dynamic models will improve the design and generalizability of iSTART-2.

Globally, this project will contribute to the AIED and EDM fields. User models are an important and often crucial aspect of ITS development. However, very few systems (if any) use dynamic data mining techniques to inform their student models. This work will be among the first studies to use techniques such as Hurst exponent analysis in real-time to inform user models that will ultimately be used to adapt the content and feedback presented to students. The methods presented here are generalizable and thus can be used in a variety of settings beyond iSTART-2. Although the goal of the current work is to design user models for the iSTART-2 system, this work is driven by the overarching goal of gaining a better understanding of students' learning processes.

3. Previous Work

My previous research has revealed that dynamic methodologies are useful tools for quantifying students' behavioral patterns within iSTART-2 [8,9,13,14,15]. For instance, Entropy is a dynamical methodology used to measure the amount of predictability within a system or time series [16]. My colleagues and I have employed post hoc Entropy analysis to quantify variations in students' behaviors within iSTART-2 and related them to performance differences. Based on students' choices within games, an Entropy score can be calculated that is indicative of the degree to which students' choice patterns are controlled versus random. In [13], students' Entropy scores were included within a regression analysis to examine how students' choices within the system influenced their self-explanation performance. Students who engaged in more controlled interaction patterns (i.e., strategic and planned out) within iSTART-2 also generated higher quality self-explanations compared to students who acted in more random or impulsive manners.

While Entropy provides an overall view of students' choice patterns within a system, it does not capture fine-grained fluctuations that manifest over time. To address this issue, Hurst exponents have been conducted using iSTART-2 log data. Hurst exponents [17] are similar to Entropy analyses in that they quantify tendencies or fluctuations present within a time series. However, Hurst exponents also act as long-term correlations that can characterize the fluctuations that manifest across time. Hurst exponents classify these fluctuations as persistent, random, or antipersistent [18]. Using this approach, we can identify when students choose to perform the same action(s) repetitively [8]. This technique affords a fine-grained look at students' behaviors across time. Although Entropy and Hurst exponent analyses have shed light upon the effects of students' interactions within an ITS on learning, the analyses thus far have all been conducted post hoc (i.e., using data mining techniques). Thus, the current work seeks to build upon these dynamical analyses and apply dynamic data mining techniques in *real-time* as a means to inform student models within iSTART-2.

4. Advice Sought

For this doctoral consortium, advice is sought regarding two core concerns. First, *what features should be included in dynamic user models?* Currently, I have solely focused on students' behaviors and in-system performance within the game-based practice portion of the system. However, iSTART-2 has powerful logging functionality capable of collecting everything from mouse movements to keystrokes. Thus, in this setting I would benefit from expert opinions or discussions concerning what features should (or could) be included within dynamic user models.

Second, *what other dynamic methodologies and tools are available and relevant to user modeling?* Thus far, I have used random walks, Entropy and Hurst analyses. However each of these measures have one or more weaknesses. For instance, to reliably calculate a Hurst exponent, multiple data points are needed (e.g., over 100), therefore calculating Hurst in real-time may not be practical in all situations (i. e., a single session study). Thus, I would benefit from expert opinion and guidance regarding other dynamic measures or methodologies that could be used in real-time as a way to inform user models within iSTART-2.

5. ACKNOWLEDGMENTS

Prior research was supported in part by the Institute for Educational Sciences (IES R305G020018-02; R305G040046, R305A080589) and National Science Foundation (NSF REC0241144; IIS-0735682). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the

IES or NSF. I would also like to thank Danielle McNamara, Rod Roscoe, Tanner Jackson, and Matthew Jacovina for their contributions to this line of work.

6. REFERENCES

[1] Murray, T. 1999. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, *10,* (1999), 98-129.

[2] Shute, V. J., and Psotka, J. 1994. *Intelligent Tutoring Systems: Past, Present, and Future* (No. AL/HR-TP-1994-0005). ARMSTRONG LAB BROOKS AFB TX HUMAN RESOURCES DIRECTORATE.

[3] Johnson, W. L., and Valente, A. 2009. Tactical Language and Culture Training Systems: using AI to teach foreign languages and cultures. *AI Magazine*, *30*, (2009) 72.

[4] Lynch, C., Ashley, K., Aleven, V., and Pinkwart, N. 2006. Defining ill-defined domains; a literature survey. In Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (Jhongli, Taiwan, June 26-30, 2006) Lecture Notes in Computer Science / Programming and Software Engineering pp. 1-10.

[5] Siemer, J., and Angelides, M. C. 1995, December. Evaluating intelligent tutoring with gaming-simulations. In *Proceedings of the 27th conference on Winter simulation* (Arlington, VA, December 03 - 06, 1995), IEEE Computer Society, pp. 1376-1383.

[6] Aleven, V. A., and Koedinger, K. R. 2002. An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive science*, *26* (2002), 147-179.

[7] Kay, J., Halin, Z., Ottomann, T., and Razak, Z. 1997. Learner know thyself: Student models to give learner control and responsibility. In *Proceedings of International Conference on Computers in Education* (pp. 17-24).

[8] Snow, E. L., Allen L. K., Russell, D. G., and McNamara, D. S. 2014. Who's in control?: Categorizing nuanced patterns of behaviors within a game-based intelligent tutoring system. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), Proce*edings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.

[9] Snow, E. L., Jacovina, M. E., Allen, L. K., Dai, J., and McNamara, D. S. 2014. Entropy: A stealth assessment of agency in learning environments. In J. Stamper, Z. Pardos, M. Mavrikis, B. M. McLaren (eds.), Proce*edings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.

[10] McNamara, D. S., Levinstein, I. B., and Boonthum, C. 2004. iSTART: Interactive strategy trainer for active reading and thinking. *Behavioral Research Methods, Instruments, and Computers*, 36, (2004), 222-233.

[11] Jackson, G. T., and McNamara, D. S. 2013. Motivation and Performance in a game-based intelligent tutoring system. *Journal of Educational Psychology, 105*, (2013), 1036-1049.

[12] McNamara, D.S., O'Reilly, T., Best, R., and Ozuru, Y. 2006. Improving adolescent students' reading comprehension with iSTART. *Journal of Educational Computing Research, 34*, (2006), 147-171

[13] Snow, E. L., Allen, L. K., Jacovina, M. E., and McNamara, D. S. 2015. Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers and Education, 26*, (2015), 378-392.

[14] Snow, E. L., Likens, A., Jackson, G. T., and McNamara, D. S. 2013. Students' walk through tutoring: Using a random walk analysis to profile students. In S. K. D'Mello, R. A. Calvo, and A. Olney (Eds*.), Proceedings of the 6th International Conference on Educational Data Mining,* (Memphis, Tennessee, July 6 -9, 2013), Springer Berlin Heidelberg, 276-279.

[15] Snow, E. L., Jackson, G. T., and McNamara, D. S. 2014. Emergent behaviors in computer-based learning environments: Computational signals of catching up. *Computers in Human Behavior, 41*, (2014), 62-70.

[16] Grossman, E. R. F. W. 1953. Entropy and choice time: The effect of frequency unbalance on choice-response. Quarterly *Journal of Experimental Psychology, 5*(1953), 41-51.

[17] Hurst, H. E. 1951. Long-term storage capacity of reservoirs. Transactions of the American Society of Civil Engineers, 116, (1951), 770-808.

[18] Mandelbrot, B. B. 1982. The fractal geometry of nature. New York: Freeman.

Use of Time Information in Models behind Adaptive System for Building Fluency in Mathematics

Jiří Řihák
Masaryk University Brno
thran@mail.muni.cz

ABSTRACT

In this work we introduce the system for adaptive practice of foundations of mathematics. Adaptivity of the system is primarily provided by selection of suitable tasks, which uses information from a domain model and a student model. The domain model does not use prerequisites but works with splitting skills to more concrete sub-skills. The student model builds on variation of Elo rating system which provide good accuracy and easy application in online system. The main feature of the student model is use of response times which can carry useful information about mastery.

1. INTRODUCTION

Our aim is to develop a practice system focused on basic mathematics which uses concepts of Computerized adaptive practice [8], i.e. to provide children with tasks that are most useful to them. We focus especially on detecting mastery and fluency using both correctness and timing information about children's responses.

Mathematics is usually associated with procedural knowledge. However, for achieving mastery of advanced topics it is necessary to solve some basic mathematical tasks at the level of fluency and automaticity. Good example of this is multiplication of small numbers which starts as procedural knowledge (child knows that $3 \cdot 5$ is $5 + 5 + 5$ and is able to complete calculation) but ends as declarative knowledge (child knows $3 \cdot 5$ is 15 without further thoughts) [15]. In both cases child gives correct response with high probability and the system is not able to distinguish between these scenarios based only on the correctness of the answer. Thus we want incorporate into our student model the information about response time, which is necessary to detect mastery, the state when the child is correct and fast.

Because our goal is to lead a child to automaticity we want to analyse strengths and weaknesses of the child at the level of individual items. Thus we need to track child's skills in great detail and we treat every item in the system independently. Also the fact that various graphical representations of the same task influence difficulty of the item, highlight need to track their difficulty individually. To estimate correctly difficulties of the items requires a lot of expertise, it is time consuming and is not always reliable. Therefore we do not want to make any assumptions about difficulties of the items and we rather use model which can estimate the difficulty of the solving data from the system. As a consequence we will be able to easily analyse which items are more difficult and why.

Proposed system is called MatMat and is currently available online in beta version at `matmat.cz` for all children (the system is so far implemented only in Czech) and it is free to use. The goal of the system is to provide adaptive practice of arithmetic operations which guide children from basic work with numbers (e.g. counting objects) to mastery of basic mathematical operations.

In contrast with complex intelligent systems for learning mathematics as Carnegie Learning's Tutors [14, 9] or ASSISTments [4] we focus only on small part of learning mathematics and we work only with atomic tasks. Therefore the system does not work with explanations of curriculum or hints and focuses on adaptive selection of tasks and appropriate feedback. Between related systems belongs Dybuster Calcularis [6] which works with basic math especially in context of dyscalculia; Math Garden [8] which has similar focus, works with similar student model and also incorporates time information; or FASTT Math [3] which also focus on building computational fluency.

2. MODELS

In this section we describe working draft of the domain model, which describes how is the content of the system organized, and the student model, which is built on the domain model and provides information about children who interact with the system. We have several requirements for the design of our models. We are in the situation when we use models in online environment and we rely more on collected data instead of expertise or other outside information. Hence we require models which can work on the fly and can quickly adapt to new data in the system. The goal of the student model is to provide estimation of child's abilities which are used for creation of feedback and selection of suitable tasks to practice.

2.1 Domain Model

Mathematics is very complex domain full of diverse components and relationships. Even in our very simplified case, when we considered only basics, situation can still be relatively complicated. One way how to build a domain model for mathematics is based on Knowledge space theory [1]. This approach splits the curriculum to skills a defines relations of prerequisites between them. This oriented graph can then be treated as dynamic Bayesian network [7].

We used different approach which allows us to capture information about very specific abilities, e.g. how good is child in multiplication of 5 and 7. The relations between such concrete skills, are not always prerequisites, e.g. the abilities to compute $5 \cdot 7$ and $5 \cdot 9$ are not one prerequisite to another but they are clearly dependent. We organized the skills into the tree structure (Figure 1) where every node corresponds to skill and its successors to more concrete sub-skills. Similarity of skills then can be expressed as level of the nearest common ancestor. Denote the fact that a skill d is ancestor of a skill c as $d > c$.

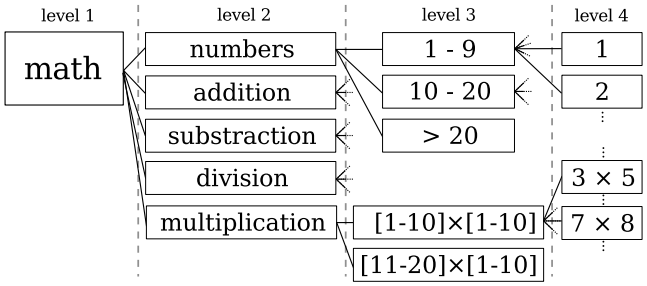


Figure 1: The tree structure of the skills

The root of the tree is a global skill which represents overall knowledge of mathematics. Under that are skills which correspond to basic units in system (level-2) — numbers, addition, subtraction, multiplication and division. In level-3 are sub-skills which represent concepts (inspired by [6]) within parent skill, e.g. under 'numbers' skill are 'numbers in range from 1 to 9', 'numbers in range from 10 to 20', 'numbers greater then 20', ...; or under 'addition' are 'addition in range from 1 to 9 (without bridging to 10)', 'addition in range from 10 to 20 with bridging to 10', ... And finally level-4 skills correspond to the tasks for which mastery on the level of declarative knowledge is expected. Example of these are skills that correspond to numbers (1, 2, 3, ...), simple addition tasks ($1+2$, $5+7$) or multiplication of numbers smaller than 10 ($3 \cdot 5$, $7 \cdot 8$). There are no level-4 skills for more complicated task (e.g. $11 \cdot 13$) for which procedural knowledge is more involved. The items representing these tasks belong typically to more general level-3 skills.

In current model every item in the system is mapped to exactly one skill (typically a leaf skill). So under a skill are multiple items. In case of the more general level-3 skills it can be tens or hundreds. In case of the level-4 skills there are from 2 to 10 items which are various forms of the task ($5+7$ and $7+5$) and different graphical representations of task (numbers, objects, number line ...).

2.2 Student Model

Rather then the discrete representation of ability (known or unknown) we used the continues representation, which is more suitable for our situation when we need to track abilities also for relatively general skills. The relation between these abilities and expected probability of correct answer is defined by a logistic function.

For the skill from s and the child c model estimates the value v_{sc} which represents difference of ability relative to parent skill. Overall value of ability is then $\theta_{sc} = \sum_{s \leq \bar{s}} v_{\bar{s}c}$. This approach allows to capture relations between leaf skills. Information obtained from observation about one ability can be naturally propagated to other related abilities. This is especially important for new children in the system with small number of responses (relatively to large number of abilities). The model also estimates the difficulties β_i of the items i, which can be interpreted as a required ability to have 50% chance of solving item correctly. Expected response is then $e_{ci} = \frac{1}{1+e^{\beta_i - \theta_{sc}}}$.

To estimate abilities and difficulties we used a model based on Elo rating system [2] and PFA [12] which is inspired by models which have been successfully used in other projects [8, 11]. The main idea is to update all related abilities and item difficulty based on unexpectedness of response after every answer. To empathize the fact that the correct answer (even repetitive) does not mean mastery we need to take into account the response time t_{ci}. This can be achieved by extension of discrete response r_{ci} (correct or incorrect) to continuous one where values between 0 and 1 mean the correct answer but with longer time than the targeted time τ_i. Example of this extension is decay of the response value exponentially relatively to the ratio of t_{ci} and τ_i (Figure 2).

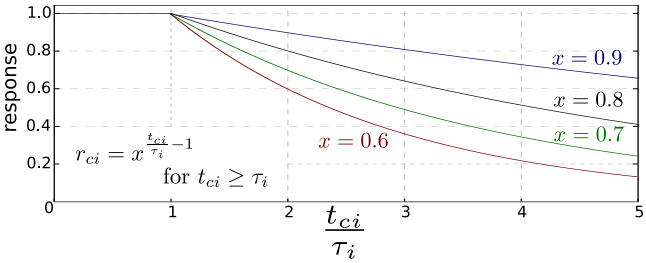


Figure 2: The response value for the correct answers

After the answer, all abilities θ_{sc} belonging to ancestors' skills s are updated. Updates of abilities are performed sequentially from the root of the skill tree. If the answer is the first answer of the child to the item, the difficulty of item β_i is also updated.

$$\beta_i = \beta_i + \frac{\alpha}{1 + \beta \cdot n_i} \cdot (e_{ci} - r_{ci}),$$
$$\theta_{sc} = \theta_{sc} + \gamma_s \cdot K_r \cdot (r_{ci} - e_{ci}).$$

The parameters α and β define shape of the decay function [13] which prevents excessive influence of recent responses. The decay function takes argument n_i – number of previous updates of that difficulty. Parameter K_r corresponds to PFA updates and depends on correctness of answer. Parameter $\gamma_s \in [0, 1]$ tells how much response to the item testifies about

a skill and consequently, how much information obtained from response is propagated to sub-skills. Reasonable values of γ_s are near 1 for the most concrete skills and near 0 for the global skill.

2.3 Item Selection

The selection of an appropriate item that suits the ability of a child is a key feature of the system and has to balance several aspects. The system should not select the same or similar item in a short time, it should select diverse items for better exploration of child's abilities and, foremost, the system should select items with appropriate difficulty – not already mastered (high probability of success) and not too difficult (small probability of success). Currently used algorithm is very similar to the one described in [11]. Only difference is in bringing into account also similarity of items (e.g. $5 + 7$ is similar with $7 + 5$).

3. FUTURE WORK

Most of adaptive educational systems currently work only with correctness of responses. Our goal is to find out if this classical approach can be robustly extended by taking into account timing information and if this extension can be useful in building fluency in the basic mathematical tasks. To target this questions we proposed the system described in this work. This system is still in testing phase but the first analysis of 28 thousand collected answers, show that the ability and difficulty values estimated by the student model make intuitive sense, the system can adapt quickly and the item selection algorithm works reasonably. However, there is a lot of space for improvement.

The domain model can be enriched with prerequisites which can be useful for both ability estimation and for item selection. The current choice of the skills used in the domain model should be reviewed by a domain expert or compared with automatic methods which use collected data [10]. The proposed student model is incorporating response time but current approach is quite simplified and explicitly does not distinguish between accuracy and speed, which can be modeled separately. Also it is not clear how to set, or rather automatically estimate, targeted response times τ_i. Next characteristic of the model is propagation of information about abilities across all skills, which is useful in first phases but later can be undesirable. The propagation is closely connected to parameters γ_s and their influence to the model behaviour should be investigated.

To evaluate our approach the proposed models will be compared to alternative models (e.g. Bayesian network model [7] which works with prerequisites) or simpler versions of Elo model (e.g. model which uses only one global skill and independent local skills [11]). The comparison of the models can be done offline with respect to the quality of predictions or online by comparison of an improvement rate or behaviour of children groups using different models and item selection strategies. These comparisons should bring some light an whether the proposed methods are useful.

Acknowledgements

Author thanks Juraj Nižnan for cooperation on the described system, Radek Pelánek for guidance and useful comments and Jan Papoušek for fruitful discussions.

4. REFERENCES

[1] Jean-Paul Doignon and Jean-Claude Falmagne. Spaces for the assessment of knowledge. *International journal of man-machine studies*, 23(2):175–196, 1985.

[2] A. E. Elo. *The rating of chess players, past and present, volume 3*. Batsford London, 1978.

[3] Ted S Hasselbring, Alan C Lott, and Janet M Zydney. Technology-supported math instruction for students with disabilities: Two decades of research and development. *Retrieved December*, 12:2005, 2005.

[4] Neil T Heffernan and Cristina Lindquist Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[5] Tanja Käser, Gian-Marco Baschera, Juliane Kohn, Karin Kucian, Verena Richtmann, Ursina Grond, Markus Gross, and Michael von Aster. Design and evaluation of the computer-based training program calcularis for enhancing numerical cognition. *Frontiers in psychology*, 4, 2013.

[6] Tanja Käser, Alberto Giovanni Busetto, Gian-Marco Baschera, Juliane Kohn, Karin Kucian, Michael von Aster, and Markus Gross. Modelling and optimizing the process of learning mathematics. In *Intelligent Tutoring Systems*, pages 389–398. Springer, 2012.

[7] S Klinkenberg, M Straatemeier, and HLJ Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[8] Kenneth R Koeclinger, Albert T Corbett, and Steven Ritter. Carnegie learning's cognitive tutor: Summary research results. 2000.

[9] Juraj Nižnan, Radek Pelánek, and Jiří Řihák. Mapping problems to skills combining expert opinion and student data. In *Mathematical and Engineering Methods in Computer Science*, pages 113–124. Springer, 2014.

[10] Jan Papoušek, Radek Pelánek, and Vít Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining (EDM)*, pages 6–13, 2014.

[11] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[12] Radek Pelánek. Time decay functions and elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.

[13] Steve Ritter. The research behind the carnegie learning math series. *Online]. Disponible en: http://www. carnegielearning. com*, 2011.

[14] Duane F Shell, David W Brooks, Guy Trainin, Kathleen M Wilson, Douglas F Kauffman, and Lynne M Herr. *The unified learning model: How motivational, cognitive, and neurobiological sciences inform best teaching practices*. Springer Science & Business Media, 2009.

Doctoral Consortium: Integrating Learning Styles into adaptive e-learning system.

Huong May Truong
Eduworks Initial Training Network
Corvinno Technology Transfer Centre, Budapest, Hungary
mtruong@corvinno.com

ABSTRACT

This paper provides an overview and update on my PhD research project which focuses on integrating learning styles into adaptive e-learning system. The project, firstly, aims to develop a system to classify students' learning styles through their online learning behaviour. This will be followed by a study on the complex relationship between learning styles, learning supports and learning outcomes. The findings can contribute significantly to the area that is still left with several unanswered questions. In addition, based on the results, meaningful recommendations and suitable online adaptation can also be made to a wide range of stakeholders of the education system.

Keywords

Learning Styles; E-learning; Adaptive learning system; Data mining; Learning analytics.

1. INTRODUCTION

Learning styles which are defined as students' preferred ways to learn can play an important role in the development of the e-learning system. With the knowledge of different styles, the system can offer insights and advices to a wide range of stakeholders such as students and teachers to effectively organise their learning materials and studying activities to optimise the learning paths. For example, under Felder-Silverman's learning styles frameworks [5], students may prefer to process information actively or reflectively. For "active" students, they perform better through interaction with other students compared to the traditional classroom. Thus, it is advisable for teachers to provide such group the opportunity to interact and discuss the learning topic [5] A recent report by Thalmann [17] surveying e-learning system developers even suggested that learning styles were the most useful personalization sources among other factors such as background knowledge and user history. In addition, there are clear potential benefits for both fields of learning styles research and e-learning system development. On one hand, the integration can help to improve the e-learning experience, providing means to build rules for personalising resources. On the other hand, the e-learning system which allows data mining and computerized algorithms can offer opportunity to observe, analyse and gain further information into students' learning styles throughout the whole learning process which could not easily be done in traditional learning styles theories research.

Nevertheless, integrating the traditional theories which have the base in psychology, pedagogy and cognitive research into the online environment is not a straight forward task. Measurement methods provided by traditional theories are mostly based on long self-judgment questionnaires [4] and thus, do not provide sufficient means fitting to the e-learning system. Furthermore, scholars still do not agree on how to optimize the matching process between learning styles and learning supports [4, 16] which leaves places for further exploration.

With the motivation to address these research problems of integrating learning styles into adaptive e-learning system, this paper contains my proposals as well as the current research progress.

2. PROBLEM STATEMENT AND PROPOSED CONTRIBUTIONS

2.1 Research Questions

In a more comprehensive way, learning styles, according to Keefe [11], can be defined as: "The composite of characteristic cognitive, affective, and physiological factors that serve as relatively stable indicators of how a learner perceives, interacts with, and responds to the learning environment". On the traditional theories side, which is mainly based on psychological, pedagogical and cognitive research, the review by Coffield, Moseley, Hall, and Ecclestone, [4] has identified over 70 theories and models. While there are no theories that outperform others [4], theories that consider the flexibility and changes of styles overtime appear to be more popular in e-learning application. Notable theories in this group include: Felder-Silverman's learning styles theory [5] which divides learners based on their: information input, information process, perception, and understanding, Kolb's Learning styles inventory [12] and Honey and Mumford's Learning styles [10] which both divide styles based on their proposed learning cycles.

The theories undoubtedly provide an essential foundation for learning style research. Nevertheless, there are several unsettled issues when applying to the online environment. In this proposal, with the aim to integrate learning styles into e-learning systems, I focus on two main ones: a) learning style classification system in e-learning and b) the relationship between learning styles, learning support and learning outcomes.

2.1.1 Learning Styles Classification

In terms of learning styles measurements, a review by [4] shows that almost all of the theories are assessed by questionnaires or surveys, requiring learners to evaluate or rank their own styles and behaviours. This type of qualitative measurement suffers many downsides. Firstly, it relies on students' self-judgments which can be bias. Secondly, although learning styles, according to many theories, can change over time, surveys and questionnaires only measure styles at one point in time. Several surveys are, in addition,

questioned by critics in terms of validity and reliability [4]. It is time consuming as there are surveys that can reach over 40 - question long (e.g. [16, 23]), and as a result, they may not be updated easily. Hence, these disadvantages of a long, time consuming, and self-judgement-based measurement create several difficulties when it comes to the adaptive e-learning system development.

In recent years, the application of machine learning which allows computerized algorithms to quickly analyse and mine huge online behaviour dataset provides the opportunity to develop new measurement methods that overcome the current drawbacks. As a result, it has opened a call for integrating learning styles with e-learning system using machine learning application [1, 14].

With the area is still at its early stage, there is still only a few proper peer-reviewed researches that attempt to tackle this theories integration issue [1]. Numerous problems remain unanswered. Firstly, several learning styles predictors can be traced in previous literature which show a complex relationship between learning styles and online behaviour. For example, to measure learning styles under Felder-Silverman's framework, while [6] used attributes related to forums, chats, exam revision etc., [20] measured using variables related to assessment such as questions answering time, performance on the test, questions checking time etc. Nevertheless, through my literature review of 51 previous papers [18], none of the papers has managed to compare the power of different predictors. The results of such comparisons will very interesting and valuable as it can act as guidelines for future developers and contribute significantly in improving the performance and efficiency of classification models.

Secondly, in terms of machine learning classification algorithms, among 51 papers reviewed [18], the most popular method identified is Bayesian networks (and Naïve Bayes – a special case of Bayesian network) (e.g.[6, 7],) which has the base in Bayes theorem. This type of approach has shown positive results in a number of researches so far. Nevertheless, for Bayes theorem to work, it requires a number of conditional probabilities and the relation network to be identified which are not always straight forward tasks. Another popular branch of methods is rules based (e.g. [7, 20]). This group of methods is interpretable, however, it relies heavily on how well the researchers "translate" the theory into the online world. For example, Graf et al., [8] based on the description of learning styles from Felder and Silverman's to obtain "rules" e.g. If a student used exercise more frequently, he is more likely to prefer active learning style. The remaining group of researchers still focuses mainly on single supervised methods which left places for the application of other advanced machine learning methods such as hybrid and ensemble machine learning that combine different machine learning algorithms together. Such advanced methods have shown significant higher performance than single algorithm in other applications such as medical and finance ([3, 19]).

Finally, current models also lack generalisation ([2, 15]). Researches are still employed to only one particular context. Akbult and Cardak [1] pointed out that the research population for almost all of the researches is still limited to undergraduate students. Thus, it raises the question if such models can be applied to a different situation from their own.

These open gaps for a better classification model found in learning styles research field have led to the following research questions:

- How can we incorporate machine learning and traditional learning styles theories? How can we measure learning styles through online behaviour?
- Which predictors are the most meaningful in predicting learning styles in online environment? What is the relationship between online behaviour and learning styles?
- What is a more effective way for learning styles classification compared to current approaches?
- Is it possible to generalize the measurement method?

2.1.2 The relationship between learning styles, learning support and learning outcomes.

The second issue relates to the relationship between learning styles, learning supports methods and learning outcomes. On one hand, students with different learning styles prefer to study in different ways. On the other hand, researchers still do not agree on how to optimise this matching process between learning styles and learning supports and interventions ([4, 16],). At the same time, the relationship between learning styles and learning outcomes is still unclear [1]. Pashler, McDaniel, Rohrer and Bjork [13] reported that previous researches still show flaws in their methodology, which as the result, fail to persuasively show the effect of learning instruments on students with different learning styles. There are also several contradictory results. For example, Ford and Chen (2001 cited in [4]) suggested that matching students learning styles with their preferred teaching style is associated with better learning results. However, Holodnaya [9] found that it will be beneficial to study under a mismatched condition. Consequently, to be able to provide reliable feedback to different stakeholders of the education system, it is essential to revisit the issue. The following research questions have been raised:

- How can we match learning supports to learning styles to improve learning outcomes?
- Under the same condition, are learning styles making any differences to learning outcomes? Are there any styles that are more preferable under certain circumstances?

2.2 Potential contributions

Overall, the area of integrating learning styles theories into e-learning systems has gained interest over the past years, yet there are still many questions that are underexplored. This research, thus, firstly, will address a number of research gaps in the field such as the relationship and influence of different online attributes on learning styles. Interesting patterns between different styles and behaviours can, as the consequence, be identified. Secondly, it aims to advance in the methods for learning styles classification which will improve the accuracy and efficiency. Lastly, it will reconfirm the debate in terms of the relationship between learning styles, learning outcomes and learning supports that can contribute significantly in helping the students to excel in their study. In addition, the findings can also work as guidelines and contribute for future e-learning development research.

3. PROPOSED METHODS AND CURRENT PROGRESS

The research will be carried out in 2 phases that each dedicates to a problem mentioned in section 2. At the current stage, I focus on phase 1 which is to develop a learning styles classification system. Thus, this section will centre mainly on phase 1's method and updates.

To develop a classification method, the following process will be carried out: it will start off with learning style theories selection, then attributes selection and finally, classification methods development and evaluation.

Firstly, while the learning styles classification field is crowed [4], through careful review in terms of theories reliability, validity, usefulness in recommendation, in this study, I chose to follow Felder-Silverman theory which is one of the most popular theories implemented in e-learning system [1]. Hence, it will also provide the opportunity for performance benchmarking.

In terms of attributes selection, I have carried out a literature-based survey [18] focusing on not only previous personalization system development researches, but also papers studying the relationship of learning styles and online behaviour. The result is a long list of potential attributes (over 80 items) which can be divided into three main sources including static data such as user background, ethnics, major etc., online behaviour e.g. time spent on certain activities and other personalization sources e.g. intelligence, memory capacity.

The data for different attributes is currently being programed and collecting for classification methods development using a learning system developed at Corvinno called STUDIO. Felder-Silverman's ILS survey has also been carried out as it is still the base line for online modelling evaluation that has been used in almost all of the previous papers. Over 250 undergraduate students are being observed with the plan of collecting data on the second group of students for model generalisation evaluation ability in the next school term in September.

Lastly, the classification methods development is still in the early stage. As most of previous researches still use single classification methods, I see an opportunity to apply more advanced techniques such as ensemble machine learning which combines different single algorithms to improve the performance. This branch of methods has shown to outperform single methods in other applications such as medical and finance.

4. FUTURE DIRECTION AND ADVICES SOUGHT

The research is still at the early stage and thus, there are a number of challenges ahead that I hope the consortium can provide advices on or sharing similar experiments and insights related to:

- Attributes comparison in the case with huge number of attributes and algorithms tested.
- While I will focus on ensemble and hybrid methods, I am also interested in if there is any other method, especially in the area of sequence mining.
- Generalisation: Is this necessary/possible to generalise the detection models? What are the conditions that we have to test for generalisation? Is testing on different populations enough?

5. ACKNOWLEDGMENT

This work is being carried out under the framework of Eduworks Initial Training Network, Marie Skłodowska-Curie actions (MSCA) FP7 of the European Commissions.

6. REFERENCES

[1] Akbulut, Y. and Cardak, C.S. 2012. Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011. *Computers & Education*. 58, 2 (2012), 835–842.

[2] D Baker, R.S. 2010. Mining data for student models. *Advances in intelligent tutoring systems*. Springer. 323–337.

[3] Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J.C. 2011. Data mining for credit card fraud: A comparative study. *Decision Support Systems*. 50, 3 (2011), 602–613.

[4] Coffield, F., Moseley, D., Hall, E. and Ecclestone, K. 2004. Learning styles and pedagogy in post-16 learning: A systematic and critical review. *London: Learning and Skills Research Centre.* (2004).

[5] Felder, R.M. and Silverman, L.K. 1988. Learning and teaching styles in engineering education. *Engineering education*. 78, 7 (1988), 674–681.

[6] García, P., Amandi, A., Schiaffino, S. and Campo, M. 2007. Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education*. 49, 3 (2007), 794–808.

[7] Graf, S., Kinshuk, K.D. and Liu, T.-C. 2008. Identifying Learning Styles in Learning Management Systems by Using Indications from Students' Behaviour. *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on* (2008), 482–486.

[8] Graf, S., Kinshuk, K.D. and Liu, T.-C. 2009. Supporting Teachers in Identifying Students' Learning Styles in Learning Management Systems: An Automatic Student Modelling Approach. *Educational Technology & Society*. 12, 4 (2009), 3–14.

[9] Holodnaya, M.A. 2002. Cognitive styles: on the nature of individual mind. *Per Se, Moscow*. (2002).

[10] Honey, P. and Mumford, A. 1986. *Using your learning styles*. Peter Honey Maidenhead,, UK.

[11] Keefe, J.W. 1979. Learning style: An overview. *Student learning styles: Diagnosing and prescribing programs*. (1979), 1–17.

[12] Kolb, D.A. 1981. Learning styles and disciplinary differences. *The modern American college*. (1981), 232–255.

[13] Pashler, H., McDaniel, M., Rohrer, D. and Bjork, R. 2008. Learning styles concepts and evidence. *Psychological science in the public interest*. 9, 3 (2008), 105–119.

[14] Romero, C., López, M.-I., Luna, J.-M. and Ventura, S. 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*. 68, (2013), 458–472.

[15] Scheuer, O. and McLaren, B.M. 2012. Educational data mining. *Encyclopedia of the Sciences of Learning*. Springer. 1075–1079.

[16] Stash, N., Cristea, A.I. and De Bra, P. 2006. Adaptation to learning styles in e-learning: Approach evaluation. (Honolulu, Hawaii, 2006).

[17] Thalmann, S. 2014. Adaptation criteria for the personalised delivery of learning materials: A multi-stage empirical investigation. *Australasian Journal of Educational Technology*. 30, 1 (2014).

[18] Truong, H.M. 2015. Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in Human Behavior*. (2015), doi: 10.1016/j.chb.2015.02.014 .

[19] Wang, G., Hao, J., Ma, J. and Jiang, H. 2011. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*. 38, 1 (2011), 223–230.

[20] Wen, D., Graf, S., Lan, C.H., Anderson, T. and Kinshuk, K.D. 2007. Supporting web-based learning through adaptive assessment. *FormaMente Journal*. 2, 1-2 (2007), 45–79.

Modeling Speed-Accuracy Tradeoff in Adaptive System for Practicing Estimation

Juraj Nižnan
Masaryk University Brno
niznan@mail.muni.cz

ABSTRACT

Estimation is useful in situations where an exact answer is not as important as a quick answer that is good enough. A web-based adaptive system for practicing estimates is currently being developed. We propose a simple model for estimating student's latent skill of estimation. This model combines a continuous measure of correctness and response-times. The advantage of the model is its simple update method which makes it directly applicable in the developed adaptive system.

1. INTRODUCTION

Estimation is a very useful skill to possess. Particularly in situations where an exact answer is not as important as being able to quickly come up with an answer that is good enough (e.g., total amount on a bill in a restaurant, number of people in a room, total of the coins in a wallet, number of cans of paint needed for painting a room, converting between metric and imperial units). It was shown that estimation ability correlates with the ability to solve computational problems [2, 9, 8]. Because estimation is so useful, we have decided to develop a computerized adaptive system that will let its users practice estimating by solving various tasks.

The adaptive system will include exercises for practicing numerical estimation (results of basic arithmetic operations, converting between imperial and metric units, converting between temperature units, currencies and exchange rates) and visual estimation (counting the number of objects in a scene).

In order to provide adaptive behavior of the system, we need a way of inferring student's ability of estimation. In our setting, the binary-valued correctness-based modeling approach is not suitable. We do not expect the users to input exact responses, we expect them to input their best estimates. So our model should work with some measure of the quality of an answer. Another important point is the speed-accuracy tradeoff. Figure 1A shows a hypothetical tradeoff curve for one user with fixed ability. User can answer a task very quickly but it will probably be a very rough estimate. Or he/she can decide to spend more time on the task and respond with a more precise answer. Therefore, response-time should be a vital part of our model.

The system should be able to detect prior skill (i.e., how good the user was at estimation before he started using the system) which can be deduced from the first interactions of the user with the system. The goal of the developed system is to enable the user to get better at estimating. Therefore, the proposed model should also take into account user's improvement (or learning) over time. Figure 1B illustrates answers of several users on one task as red dots. Ideally, the system will help its users to learn to perform near the green mark, to be fast and accurate.

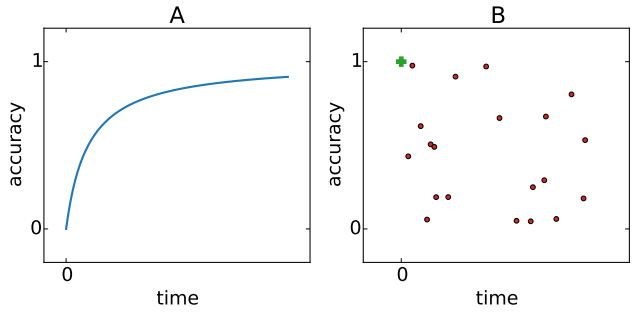


Figure 1: A) hypothetical speed-accuracy tradeoff curve, B) goal of the system

The value of the system will also be in the data that will be collected. It can be used to answer some interesting research questions. Does the speed-accuracy tradeoff curve have the same shape for converting between EUR and USD as for estimating the number of displayed objects? How do the learning curves look? Can estimation tasks in one area be learned more quickly than in another area? How close to the perfect mark can users push their performance? What is the influence of a countdown timer on user's performance? What is the appropriate level of challenge that motivates the users? The last question was addressed in [3], where the authors were trying to validate the *Inverted-U Hypothesis* (i.e., we most enjoy challenges that are neither too easy, neither too hard) on data collected from online estimation game called *Battleship Numberline*. They found out that the

easier the game was, the longer users played the game.

2. MODELS

In this section, we present a few existing models for combining correctness and response-times in Item Response Theory (IRT) and a model for tracking learning currently used in our other adaptive practice system. We then propose a simple model that could be used in the system for practicing estimates. The described models use a logistic function $\sigma(z) = (1 + e^{-z})^{-1}$. Users of the system (or students) are indexed by j. The items (or tasks, problems, questions) that the users solve are indexed by i.

2.1 Models from IRT

A typical example of an approach to the modeling of both correctness and response-times in Item Response Theory is from van der Linden [10]. The approach uses two models, one for correctness (binary) and the other one for response-times (distributed lognormally). The probability of success of a student j on item i can expressed by the 3PL model:

$$p_{ij} = c_i + (1 - c_i) \cdot \sigma(a_i(\theta_j - b_i))$$

where parameter θ_j is the skill of student j and a_i, b_i, c_i are the discrimination, difficulty and pseudo-guessing parameters for the item i. The logarithm of a response-time t_{ij} can be predicted by:

$$\ln \hat{t}_{ij} = \beta_i - \tau_j \tag{1}$$

where β_i represents the amount of labor required to solve item i and τ_j the speed of student j. The disadvantage of this model is that it does not model the speed-accuracy tradeoff explicitly.

An example of a model that directly combines binary correctness with response-time is Roskam's model [7]:

$$p_{ij} = \sigma(\theta_j + \ln t_{ij} - b_i)$$

Here, an increase in item difficulty (or decrease in student's ability) can be always compensated by spending more time on a problem. This tradeoff is called an increasing conditional accuracy function.

2.2 Model for factual knowledge

Here, we present a model that is currently used in a popular adaptive system for practicing geographical facts [4]. This model consists of two parts, one (Elo) estimates the prior knowledge of a student and the second one (PFAE) models student learning. A big advantage of this model is that it uses fast online methods of parameter estimation which makes it suitable for use in an interactive adaptive practice system.

The prior knowledge of a student is modeled by the Rasch (1PL) model. The probability that a student j answers item i correctly is modeled by the likelihood $p_{ij} = \sigma(\theta_j - b_i)$. The parameters are estimated using Elo rating system [1]. Elo was originally developed for rating chess players, but the process of student answering an item can be interpreted as a "match" between the student and the item. After each "match", the parameters are updated as follows:

$$\begin{aligned} \theta_j &:= \theta_j + U(n_j) \cdot (correct - p_{ij}) \\ b_i &:= b_i + U(n_i) \cdot (p_{ij} - correct) \end{aligned}$$

where $U(n)$ is the uncertainty function $U(n) = \frac{\alpha}{1+\beta n}$ and n is the number of updates of the parameter and α and β are metaparameters. The variable *correct* takes value 1 if the student has answered correctly and value 0 otherwise. This model is used for predicting– and trained on–first responses.

After the first interaction of a student j with item i has been observed, we can set student's skill in that particular item to $\theta_{ij} = \theta_j - b_i$. An extended version of Performance Factors Analysis [5] called PFAE is used to model learning and predicting the following interactions of the student with the item. Likelihood of a correct answer is $p_{ij} = \sigma(\theta_{ij})$. The update to student's knowledge of item θ_{ij} after observation is:

$$\theta_{ij} := \begin{cases} \theta_{ij} + \gamma \cdot (1 - p_{ij}) & \text{if the answer was correct} \\ \theta_{ij} + \delta \cdot p_{ij} & \text{if the answer was incorrect} \end{cases}$$

where γ and δ are metaparemeters. The reason for two different metaparameters is that the student learns also during an incorrect response.

2.3 Proposed model for estimates

Here, we propose a model that can be used in the adaptive practice system for estimates. The model combines Roskam's model and the update scheme from Elo and PFAE.

A simple extension of the correctness-based modeling to the setting of practicing estimates is to use a measure of correctness, or a *score* – a rational number ranging from 0 to 1. The way of scoring of an answer could be based on the domain being practiced by the user. For example, for the scenario where the user is estimating the number of objects in a scene, the exact answer would get a score of 1, deviating by one object a score of 0.8, etc.

The model assumes the same parameters and relationship as Roskam's model, but instead of expressing a probability of a correct answer it specifies the expected score:

$$\hat{s_{ij}} = \sigma(\theta_j + \ln t_{ij} - b_i)$$

Figure 2 shows how the score changes as a function of time for different values of user's skill θ_j (with fixed $b_i = 0$). It nicely demonstrates the speed-accuracy tradeoff.

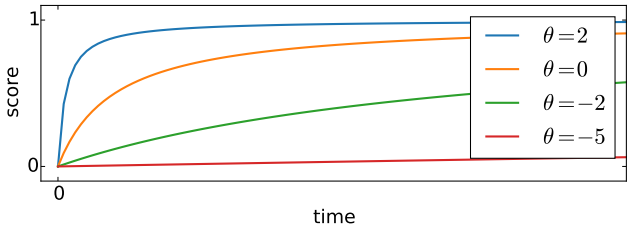


Figure 2: Score function for different values of skill

After observing score s_{ij} that user j obtained for answering item i and response-time t_{ij}, we can update model's beliefs

in the parameters:

$$\begin{aligned}\theta_j &:= \begin{cases}\theta_j + \gamma \cdot (s_{ij} - \hat{s_{ij}}) & \text{if } s_{ij} \geq \hat{s_{ij}} \\ \theta_j + \delta \cdot (\hat{s_{ij}} - s_{ij}) & \text{if } s_{ij} < \hat{s_{ij}}\end{cases} \\ b_i &:= b_i + U(n_i) \cdot (\hat{s_{ij}} - s_{ij})\end{aligned}$$

Note, that the model uses a single parameter θ_j for the student. This is different from the approach taken in PFAE, where the student has a parameter for each item θ_{ij}. While that approach is suitable for modeling the knowledge of facts – where it is reasonable to assume that the knowledge of one fact is independent of the knowledge of another – it is not suitable here. Student's ability to convert 2 miles to kilometers is surely dependent on his ability to convert 3 miles to kilometers.

We propose using separate model for each concept (e.g., estimating the number of objects, conversion lb to kg, conversion EUR to USD). It is true that student's ability to estimate items corresponding to one concept tells us something about his ability to estimate the other concepts. However, if the user does not know the conversion rate from EUR to USD then being able to estimate well the other concepts will not help him.

The model can be easily extended by adding a discrimination parameter a or a guessing parameter c (similarly to the IRT model): $\hat{s_{ij}} = c + (1-c) \cdot \sigma(a(\theta_j + \ln t_{ij} - b_i))$. These added parameters could be either metaparameters of the model or parameters of the item i. The guessing parameter may be useful for the scenario where the user has to select a value on a numberline.

As we mentioned earlier, this model suffers from the issue that increasing the time spent on an item increases the expected score. This may hold true for the instance where the user knows the underlying concept (e.g., the conversion rate from EUR to USD) but it does not hold when he does not know it. But the model uses the logarithm of response-time and the time a student is willing to spend on an item is limited. Therefore, the model should have reasonable behavior for the time interval of interest, as is demonstrated in Figure 2 by the curve corresponding to $\theta_j = -5$.

3. DISCUSSION

The model works with the response-time as a parameter. Therefore, it cannot be used for predicting response-times directly. A model similar to (1) can be used for that. Predicted time and score can be used for item selection (i.e., which item to offer the user next). This can be done by setting a target score and recommending an item with predicted score close to the target.

Does the model perform better than a simple 1PL model that does not use response-times at all? Does it make sense to add more parameters to the model? How does the model fare against more complicated models? To be able to answer these questions, we need to somehow evaluate the performance of the model. The choice of metric is interesting because a model can predict both score and response-time. When considering only the predicted score, a standard metric like RMSE can be used [6]. When we have a measure of performance, we can explore if the model is well-calibrated with respect to response-times or if the model works similarly well for all the domains (concepts).

Other question that we could ask is how well does the speed-accuracy tradeoff curve that the model assumes correspond to reality.

Acknowledgements

We thank Radek Pelánek (for guidance and useful suggestions) and Roman Orlíček (for actively working on developing the application).

4. REFERENCES

[1] A. E. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[2] S. A. Hanson and T. P. Hogan. Computational estimation skill of college students. *Journal for Research in Mathematics Education*, pages 483–499, 2000.

[3] D. Lomas, K. Patel, J. L. Forlizzi, and K. R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 89–98. ACM, 2013.

[4] J. Papoušek, R. Pelánek, and V. Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining (EDM)*, pages 6–13, 2014.

[5] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[6] R. Pelánek. A brief overview of metrics for evaluation of student models. In *Approaching Twenty Years of Knowledge Tracing Workshop*, 2014.

[7] E. Roskam. Toward a psychometric theory of intelligence. *Progress in mathematical psychology*, 1:151–174, 1987.

[8] P. M. Seethaler and L. S. Fuchs. The cognitive correlates of computational estimation skill among third-grade students. *Learning Disabilities Research & Practice*, 21(4):233–243, 2006.

[9] R. S. Siegler, C. A. Thompson, and M. Schneider. An integrated theory of whole number and fractions development. *Cognitive psychology*, 62(4):273–296, 2011.

[10] W. J. Van Der Linden. Conceptual issues in response-time modeling. *Journal of Educational Measurement*, 46(3):247–272, 2009.

Reimagining Khan Analytics for Student Coaches

Jim Cunningham
Arizona State University
Tempe, AZ
1-480-703-7394
jim.cunningham@asu.edu

ABSTRACT

In this paper, I describe preliminary work on a new research project in learning analytics at Arizona State University. In conjunction with an innovative remedial mathematics course using Khan Academy and student coaches, this study seeks to measure the effectiveness of visualized data in assisting student coaches as they help remedial math students achieve success in an online math class.

Keywords

Learning analytics, data visualization, remedial mathematics.

1. INTRODUCTION

With 77,000 students, Arizona State University has become one of the largest institution of higher learning in the United States second only to the University of Phoenix. A certain slice of newly enrolled students at Arizona State find themselves in a dilemma, they have been admitted to the university, but they have failed to meet the minimum math requirement that would allow them to start taking undergraduate classes.

These students are desirable for many reasons. They can help the university meet goals for diversity and social justice. Often these students are first generation college students. However, this population also poses challenges to the university. Being unable to meet the minimum score on the mathematics placement test not only points to gaps in a student's math background, it often points to larger issues of academic readiness.

Overcoming the hurdle of meeting the minimum math requirement has been challenging. Online remedial math classes using an adaptive learning model have historically had a pass rate of around 50%. These pass rates have remained stubbornly low despite various efforts to improve them.

2. A NEW APPROACH

In the summer of 2014, EdPlus (ASU's online education arm) decided to launch a new version of this remedial math class built around Khan Academy and undergraduate peer coaches. One of the big reasons for making this change was data. Khan, because it is a non-profit, was open to sharing the data generated by students and KA's strategies for student success. Arizona State wanted their remedial math classes to become their most data driven offering. Working with Khan had other advantages as well. Because KA has over 10 million unique users every month and over 2 billion problems worked, Khan is able to deploy and adapt its instruction at scale.

3. CHALLENGES WITH KHAN

While the strategy of building a remedial math class around Khan Academy had many strengths, there were also significant challenges. The first challenge was in the form of a problem that many schools face when working with Khan Academy. While seeking to build a comprehensive universe of math instruction, KA has developed explicit pathways to math success. Khan calls these pathways "missions." However, ASU's end goal: passing an exam that is meant to reflect the many math concepts that a student should know before entering college, cut across many Khan missions. In addition, Khan's powerful analytical tools that are meant to aid instructors in following a student's progress are tied to these missions. When math skills are being served up to students *a la carte*, as they are to accomplish ASU's remedial math program, the analytics are unraveled.

A second major challenge facing the Khan Academy program was the same challenge facing the original remedial math program. Many of these students were failing to pass the minimum math requirement to enter Arizona State University because school in general has been challenging for them for a very long time. Putting these students by themselves in an online math course of any kind could be a recipe for failure. They needed additional support. The kind of personal attention these students need is very expensive. ASU decided to control that cost by employing a system of student "coaches." Coming from a variety of majors and backgrounds, these student coaches were handpicked and given responsibility for 20-25 online students each. Their job was to monitor, guide, tutor, and encourage these students to the end goal of having their coachees pass an exam that was meant to reflect their readiness to take college level math.

In order to be effective, these coaches needed access to the data in Khan Academy about their students' progress, but because ASU's exam at the end of the remedial math course measured math skills that spanned several Khan math missions, the state-of-the-art Khan analytics that are tied to those missions were unavailable to the coaches. After a lot of work, (much of it spearheaded by the student coaches themselves) a spreadsheet was developed that was populated by weekly downloads of Khan data. It showed which math skills were practiced and which skills were mastered and matched these up to a rough metric that told the coaches whether their students were on track to successfully master all the math skills they needed before they had to take the exam.

4. RESEACH GOALS

The goal of my research is create custom data visualizations that fit ASU's mission for this remedial math class and then measure the effectiveness of these analytics in assisting the student coaches in their work of creating student success. These analytics are specifically aimed at enabling the student coaches to visualize the

large amounts of data generated through Khan Academy. Khan stores data on each student's attempt to solve a math problem related to a particular skill. There is also data on how many times a student views a Khan video on a math concept or asks for hints when attempting a math problem. Because we are re-envisioning the analytics from the ground up, we have an opportunity to create analytics that are similar to the ones that Khan has created for its missions yet improve these analytics for ASU's specific purposes and create dashboards that visualize the data in other ways that may be even more useful for the student coaches. Because the coaches only had access to Khan data through a spreadsheet the first semester the new remedial math class was taught, there is an opportunity to compare the success of coaches assisting their students with the spreadsheet data versus those using the more sophisticated data visualizations produced from the student's actions within Khan Academy.

In order to achieve this goal, ASU is teaming up with Blue Canary, a learning analytics company headquartered in Chandler, Arizona. Blue Canary and I are working directly with Khan Academy to address data flow issues including creating API's that will automatically access data from Khan databases that will be feeding the dashboards and graphics created for the student coaches. I am also going to be working with Blue Canary to create dashboards and data visualization tools for the Khan online math class in Tableau. These dashboards are directly aimed at assisting student coaches while they help their math coachees achieve success.

5. RESEARCH QUESTIONS

Once the dashboards are created and the student coaches start using them to assist their math students, we can start to address this research question: Do data visualization tools enable student coaches to better assist remedial math students entering Arizona State University achieve success?

6. RESEARCH DESIGN

The preliminary design of this study is to compare data generated by two cohorts of remedial math students. The first cohort has been guided by student coaches who have been accessing the Khan data on through a spreadsheet created to keep track of skill practice and mastery. The second cohort will be guided by student coaches who have access to the data visualization tools and dashboards created by myself and Blue Canary in Tableau. The ultimate measure of coach success will be the pass rate of their students at the end of the course. In addition, there will be many other metrics to measure, as well, such as student engagement and persistence.

This research is in the early stages. Mike Sharkey from Blue Canary and myself have been meeting with student coaches and instructional designers of the remedial math program to assess the needs of the student coaches and talk about possible data visualizations that may be helpful. API's are being designed pull data from Khan for the analytics and dashboard layouts. While we are working on this, data is being generated by students in Khan Academy who are working with coaches that are relying on the spreadsheet to access data about the progress of their students in Khan. I am currently in second year of a four year PhD program, so we have some time to make adjustments and work out problems as they arise.

Data Analysis Tools and Methods for Improving the Interaction Design in e-Learning

Paul Stefan Popescu
University of Craiova
Department of Computer Science
Craiova, Romania
+40724571133
sppopescu@gmail.com

ABSTRACT

In this digital era, learning from data gathered from different software systems may have a great impact on the quality of the interaction experience. There are two main directions that come to enhance this emerging research domain, Intelligent Data Analysis (IDA) and Human Computer Interaction (HCI). HCI specific research methodologies can be used to present the user what IDA brings after learning and analyzing user's behavior. This research plan aims to investigate how techniques and mechanisms available in both research areas can be used in order to improve learners' experiences and overall effectiveness of the e-Learning environment. The foreseen contributions relate to three levels. First is the design and implementation of new algorithms for IDA. The next level is related to design and implementation of a generic leaning analytic engine that can accommodate educational data in attempt to model data (i.e., users, assets, etc.) and provide input for the presentation layer. Last and top level is represented by the presentation layer where the output of the underlying levels adapts the user interface for students and professors.

Keywords:

Learning analytics, intelligent data analysis, interaction design, user modeling

1. INTRODUCTION

Standard books or their digital versions (eBooks) or standard e-Learning environments are usually just a simple presenting method of the learning material. In this digital era our day by day devices must became proactive to our needs, i.e. they have to know what we need before we even have to ask them. Considering the field of e-Learning, in order to find user's needs and to improve his learning experience we can log various activity related data as a first step in a data driven analytic engine. These actions may define learners' behavior in e-Learning environments providing IDA with raw data to be analyzed. Based on this data IDA creates a data model which is based on user's performed actions. A sample output of the IDA process may be represented by a user model that is aimed to directly influence the user interface.

Learning using on-line educational environments is getting more and more popular but the effectiveness of interaction between students or students and professors is usually poorer than the interaction in physical educational environments. Improving the interaction design process in e-Learning platforms may have a direct impact on the effectiveness of the learning and be achieved by following a data driven approach. The proposed approach is related to several prerequisites and the learning resource that needs to be well structured and presented. Others are related to the interaction between students and the links that can be created between them, proper data visualization techniques, interpretation of results, adequate data analysis processes with specific goals regarding interface adaptation.

2. RELATED RESEARCH IN I.D.A.

Learning analytics and Machine Learning[2] is still one of the most interesting parts of the IDA research area. One research area of this domain is related to the classification procedures. Some of them are related to the usage of classification on text[1] and some of them are regarding to usage of classification as an user analyzing method[4].

Analysis of students' activities in the online educational systems with the goal of improving their skills and experience through the learning process has been an important area of research in educational data mining. Most of the techniques are trying to predict student's performances[5,6,7,12] based on their actions.

The work in this domain started in the year of 2005 with a workshop referred to as 'Educational Data Mining' AAAI'05-EDM in Pittsburg, USA[8] which was followed by several related workshops and the establishment of an annual international conference first held in 2008 in Montreal[9] . Before of EDM, user modeling domain was the one that was encapsulating this research area.

Several papers, journals and surveys have been written but only two books were published: the first is "Data mining in E-learning"[10] which has 17 chapters oriented to Web-based educational environments and the second is "Handbook of Educational Data Mining"[11] which has 36 chapters about different types of educational settings.

In this research proposal the goal is to combine HCI with IDA and educational research in order to improve the learners experience in digital educational environments. This domain is also related to Intelligent Interfaces research area.

3. RESEARCH AND DEVELOPMENT STATUS

As research status two papers have been written so far.
I am a co-author of the paper Advanced Messaging System for On-Line Educational Environments[3]. This paper presents a method of using a classification procedure for retrieving a set of recommended messages that might be interesting to students.

The second paper is entitled „Building an Advanced Dense Classifier"[4], which has already been published at IDAIR 2014 and won the best paper award. This paper presents a classifier that implements several extra functionalities which can lead to better results. Its goal is to build a Decision Tree classifier that accommodates data (instances). This new data structure extends the functionality of a Decision Tree and is called DenseJ48. This new classifier implements efficiently several extra functionalities besides the core ones that may be used when dealing with data.

Based on this paper, as development background a Weka package which implements the classifier's functionalities is under development. I am also a contributor (http://apps.software.ucv.ro/Tesys/pages/development.php) of Tesys[13], an e-Learning platform used in several faculties from Craiova, mainly focusing on the eLeTK (e-Learning Enhancer Toolkit)[14] module. This is how I found out about Intelligent Data Analysis and Information Retrieval, and the benefits these research areas can bring to the online educational environments.

As relevant training in September 2013 I applied for and obtained a scholarship for attending the 9th European Summer School in Information Retrieval, which took place in Granada, Spain. Being part of this event helped me improve my knowledge in the domain of Information Retrieval – the presentations covered most of this research area, from basics to evaluation techniques and Natural Language Processing. Later I attended Research Methods in Human-Computer Interaction between 25th and 31th of July 2014 in Tallinn, Estonia.(http://idlab.tlu.ee/rmhci) in order to deepen my knowledge of HCI research methodologies.

4. RESEARCH PROBLEMS FROM PHD PROPOSAL

Problems related to this research can be structured in a three layer representation. There is a certain need for improving the interaction between the users (students, professors, etc.) and the system that provide them the learning experience. The research problems are related to closing the gap between classical and digital learning paradigms.

Development of new tools is fundamentally based on functionality provided by a generic learning analytic engine, among which there are: generic representation of learning analytics data of users, integration of various implementations of IDA algorithms, custom integration of interaction design process artifacts. All these three layers build up a learning analytics engine that is designed to run as a service along e-Learning environments in an attempt to improve the quality of the on-line educational system.

4.1 Layers description

4.1.1 Data Representation Layer

First layer is related to the representation of the raw data that can be gathered from the log files and the database. Our desire is to find what data (features, parameters, ranges, etc) is relevant for online learning environments. Based on this data we have to extract features that can define learning resources or those features that enable us to obtain a user representation.

4.1.2 Learning Analytics Layer

Based on the data gathered it is possible to employ different IDA algorithms in order to obtain custom built data pipelines. Experimenting at this level with different algorithms and different feature sets can lead to obtaining output information for solving different problems. Data aggregation and pipelining are the mainly used processes. The purpose of this layer is to offer to the next one data in a structured format which can be presented on the interface.

4.1.3 Presentation Layer

The presentation of the learning material is very important, leaving a mark on the mental model created by the learning resources. In this layer the HCI component of this proposal is employed.

Taking into consideration these aspects related to both domains we can say that there is a need for new tools that could be integrated within the digital learning environments in order to provide an improved learning experience that fulfills the user's needs.

4.2 Research questions & Proposed Approach

The questions that have to be addressed when we talk about research in e-Learning environments are related to the main actors that are using the on-line educational environments. Therefore, learners, teachers and administrators (which can do the data analyst job), by the generic meaning, are the ones we focus on because they are the main users of these systems. Secretaries of the learning environments only concur to configure the e-Learning environment.

The presented questions are from the business goal perspective. Answering these questions needs a close discussion about the presented underlying levels, which are the same regardless of the tackled issue, that define data driven process.

- *How IDA can be efficiently and effectively used for an on-line educational context?*

Proper usage and integration of IDA techniques can create a framework which data analysts and developers can employ for further work.

- *How can e-Learning resources be managed/aggregated in an IDA context?*

There are various types of resources that exist in on-line educational environments. Depending on how they are managed and aggregated, application developers can benefit from them.

- *Which are the common (general purpose) functionalities when dealing with educational data pipelines?*

Several functionalities exist in dealing with data but not all of them are feasible for working with educational data. In this particular case we need to find the most effective ones and adapt them to this particular case.

- *How can the student know his place among his colleagues and be motivated to study harder?*

This question is highly important from the student's perspective. Without knowing his place among his colleagues and

without having an explicit learning path, the learner will not have the indication of his final result and will not have the motivation to maximize his potential. In e-Learning environments, students do not participate together in courses, like in a regular environment, so they are unaware of their colleagues' knowledge level. In a traditional classroom, there is always a certain level of competitiveness, so each student is constantly motivated to improve himself. Therefore, an important goal is to achieve a similar scenario in the online educational environments, although it is not the only one. Besides being competitive, the students must also be engaged in helping others and in turn receive help when they are having difficulties understanding something.

- *How can the professors know where exactly do the students have problems, so they can adapt the course material?*

From the professor's point of view, being aware of his students' progress and the difficulties they encounter in understanding the material is possibly the most important requirement. Although each student is different and has his own learning curve, common points can be found and an overall perception can be formed. The professor must be able to build a mental model regarding the overall performance of his students. By doing so, he can modify and perfect in time the content of the course. Also, taking into consideration the fact that the difficulty level of the final evaluation must be consistent with the students' level of understanding of the course, the professor needs to be aware of that level so he can make the proper adjustments.

- *Which data should be logged in order to extract relevant information about the students?*

Any e-Learning environment whose goal is to integrate an intelligent component should be able to log the necessary data and extract the values of the features. Logging the needed data is a prerequisite to the data analysis process. Logging too much can create a useless load of the server but logging not enough will make impossible the features extraction.

Features are very important in IDA because they define the entity that will be analyzed. Choosing the right features are crucial in different IDA processes. A comprehensive list of features (with proper data types, range values and significance) should be available for further analysis.

5. CLOSING REMARKS

On-line educational environments are here from a long enough time. This aspect brings in front of the scientists many opportunities for improving the learning process and to lower the distance from the classical educational environments to the online ones. Many research areas concur to improve the learning process but the most relevant are the user centered ones.

There are 3 different research areas that concur to bring learners several improvements. IDA is the first one bringing data mining and machine learning algorithms and generate user models, followed by HCI, which is used to optimize the interfaces and create friendly interaction environments and finally the Educational research area is where we put in practice this work.

6. REFERENCES

[1] Vitomir Kovanovic, Srecko Joksimovic, Dragan Gasevic, Marek Hatala .:Automated Cognitive Presence Detection in Online Discussion Transcripts, LAK Workshops, 2014.

[2] Dragan Gasevic, Carolyn Penstein Rosé, George Siemens, Annika Wolff, Zdenek Zdráhal .: Learning analytics and machine learning, LAK Conference, 2014..

[3] Mihai Mocanu, Paul-Stefan Popescu, Dumitru Dan Burdescu, Marian Cristian Mihaescu,: Advanced Messaging System for On-Line Educational Environments Sesimbra, Portugal, 26-28 June 2013.

[4] Paul Stefan Popescu, Cristian Mihaescu, Mihai Mocanu, Dumitru Dan Burdescu, Building an Advanced Dense Classifier, 5th International Conference on Information, Intelligence, Systems and Applications, Chania, Crete, July 2014.

[5] Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G.: Predicting student performance: an application of data mining methods with an educational Web-based system. In: Frontiers in Education, 2003. FIE 2003 33rd Annual (Volume:1), pp. T2A--13. Westminster (2003).

[6] Baradwaj, B.K., Pal, S.: Mining Educational Data to Analyze Students' Performance. (IJACSA) International Journal of Advanced Computer Science and Applications, Volume 2, No. 6, 63--69 (2011).

[7] Márquez-Vera, C., Cano, A., Romero, C., Ventura, S.: Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. Applied Intelligence, Volume 38, Issue 3, 315--330 (2013)

[8] Beck, J, Proceedings of AAAI2005 workshop on Educational Data Mining, (2005).

[9] Baker, R.S.J.d., Barnes, T., Beck, J.E. (Eds.) Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings. Montreal, Quebec, Canada. June 20-21, 2008.

[10] C. Romero and S. Ventura, Data Mining in E-Learning, WIT Press, 2006

[11] C. Romero and S. Ventura, Handbook of Educational Data Mining, CRC Press, 2010

[12] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer and, W. F. Punch, "Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA" In Proceedings of ASEE/IEEE Frontiers in Education Conference, Boulder, CO: IEEE, 2003..

[13] Marian Cristian Mihaescu: Software Architectures of Applications Used for Enhancing On-Line Educational Environments, International Journal of Computer Science and Applications (IJCSA), vol. 10, ISSUE 1 MARCH 2013

[14] Burdescu, D.D., Mihaescu ,M.C.: Tesys: e-Learning Application Built on a Web Platform. In: Proceedings of International Joint Conference on e-Business and Telecommunications, pp. 315-318. Setubal, Portugal (2006)

Assessing the Roles of Student Engagement and Academic Emotions within Middle School Computer-Based Learning in College-Going Pathways

Maria Ofelia Z. San Pedro
Teachers College Columbia University
525 W 120th St.
New York, NY 10027
mzs2106@tc.columbia.edu

ABSTRACT

This dissertation research focuses on assessing student behavior, academic emotions, and knowledge from a middle school online learning environment, and analyzing their potential effects on decisions about going to college. Using students' longitudinal data ranging from their middle school, to high school, to postsecondary years, I leverage quantitative methodologies to investigate antecedents to college-going outcomes that can occur as early as middle school. The research first looks at whether assessments of learning, emotions and engagement from middle school computer-based curriculum are predictive at all of college-going outcomes years later. I then investigate how these middle school factors can be associated with college-going interests formed in high school, using the same assessments during middle school, together with self-report measures of interests in college when they were in high school. My dissertation then culminates in developing an overall model that examines how student interests in high school can possibly mediate between the educational experiences students have during middle school technology-enhanced learning and their eventual college-going choices. This gives a richer picture of the cognitive and motivational mechanisms that students experience throughout varied phases in their years in school.

Keywords

College Choices, Academic Emotions, Behavior, Knowledge, Social Cognitive Career Theory, Interests

1. Introduction

College enrollment and completion are key steps towards career success for many learners. However, well before this point, many students effectively drop out of the pipeline towards college quite early. According to Social Cognitive Career Theory (SCCT) [10], academic and career choices are shaped throughout middle school and high school by environment supports and barriers, where higher levels of interest emerge within contexts in which the individual has higher self-efficacy and outcome expectations, and these interests lead to the development of intentions or goals for further exposure and engagement with the activity [10]. Traditional studies also show that family background, financial resources, and prior family academic achievement have significant impacts on where students find themselves after high school. All of these factors, however, are fairly strong displays of disengagement. By the time these indicators are commonplace, students may be in such a precarious situation that many interventions may fail. In general, current models about successful access to postsecondary education may be insufficient to help educators identify which students are on track and which need further support [11]. Fine-grained assessments of student behaviors and academic emotions (emotions that students experience during learning and classroom instruction) have been found to influence learning outcomes [12, 13]. Hence, there is an argument to be made that engagement and academic emotions in middle school play an essential early role in the processes described in SCCT. In SCCT, students' initial vocational interests are modified by their self-efficacy, attitudes, and goals towards career development (i.e. college enrollment, career interest), which are themselves influenced by the student's learning and engagement when encountering the increasingly challenging content in middle school [1, 12] – as poor learning reduces self-efficacy whereas successful learning increases self-efficacy [cf. 2]. As such, student academic emotions, learning, and engagement during middle school may be indicative of their developing interests in career domains which may in turn influence their choice to attend college [6, 9].

For the reasons aforementioned, my research attempts to answer Bowers' [5] call to identify much early, less acute signals of disengagement, the sort that occur when students' engagement is still malleable enough for interventions to succeed. Specifically, I investigate antecedents to college attendance that occur during middle school, using assessments of engagement and disengagement to better understand how these factors interact so that I can develop possible paths to re-engagement before students develop more serious academic problems. The models I create and the analyses I conduct involve the context of an online learning environment, and hence, this work provides both a new perspective on the efficacy of the system and an opportunity to explore how the system and its data can be used to predict long-term educational outcomes – in the case of my dissertation research, intervention and support in keeping students on track towards the pathway to college.

2. Data and Related Methodologies

My dissertation leverages data acquired from both traditional research methods as well as methodologies from machine learning and student modeling in assessing the constructs I analyze in my data, which I then use in developing the outcome models I propose. For middle school measures, I use the ASSISTment system (ASSISTments) as my source for middle school interaction data, and assessed measures of student knowledge, academic emotions, and behavior by using individual models developed to infer them. ASSISTments is a free web-based tutoring system for middle school mathematics that assesses a student's knowledge while assisting them in learning, providing teachers with detailed reports on the skills each student knows [14]. Interaction data from the ASSISTment system were obtained for a population of middle school students who used the system at various school years, from 2004-2005 to 2008-2009. These students are drawn from urban and suburban districts who used the ASSISTment system systematically during the year. I assessed

a range of constructs from interaction data in ASSISTments, which include student knowledge estimates, student academic emotions (boredom, engaged concentration, confusion), student disengaged behaviors (off-task, gaming the system, carelessness), and other information of student usage. These form the features in our final model of college-going outcomes. Aside from educational software data, I also use survey data from the same students who used the system in middle school, consisting of information about the their attitude about the subject (mathematics) and about the system itself. These survey data were acquired around the same time they used the software in middle school.

For my high school measures of interest, students who used the system during their middle school years and who are now in high school, were administered with two surveys: the first is a short questionnaire that asked the highest level of math and science courses that the student completed in high school and asks the student what his/her educational and career plans are upon graduation. The second survey is the an CAPA survey, designed by Fred Borgen and Nancy Betz [4]. It is an online survey with Likert scale inputs from students that gauges their interest and confidence on certain domains and skills, and then assesses their overall self-efficacy and vocational interests using existing instruments.

A subset of our student sample who were expected to be in postsecondary stage of education by the time of data collection were identified for their postsecondary education status. For their college enrollment information, records were requested from the National Student Clearinghouse, with information such as whether they were enrolled in a college or not, the name of the university, date of enrollment, and college major enlisted if available. We supplemented this data with college selectivity classification of the said postsecondary institutions, taken from the Barron's College Selectivity Rating which classifies colleges into ten categories [7, 16], from most selective or 'Most Competitive' to 'Special' which consist of specialty institutions such as schools of music, culinary schools, art schools, etc. Another source of data includes survey data about post-high school academic and career achievements that was administered to this subset of students.

3. Preliminary Work

In developing an overall integrated model, I initially tested the predictive power of the middle school factors on separate postsecondary outcomes. First, I applied fine-grained models of student knowledge, student academic emotions (boredom, engaged concentration, confusion, frustration) and behavior on middle school interaction data to understand how student learning and engagement during this phase of learning can predict college enrollment. A logistic regression model was developed and can distinguish a student who will enroll in college (68.6% of the time, an above average performance for models created from "discovery with models"). In particular, boredom, confusion, and slip/carelessness are significant predictors of college enrollment both by themselves and contribute to the overall model of college enrollment. The relationships seen between boredom and college enrollment, and gaming the system and college enrollment indicate that relatively weak indicators of disengagement are associated with lower probability of college enrollment. Success within middle school mathematics is positively associated with college enrollment, a finding that aligns with studies that conceptualize high performance as a sign of college readiness [15] and models that suggest that developing aptitude predicts college attendance [8].

Next, I also modeled whether students will attend a selective college, combining data from students who used the ASSISTment system with data on college enrollment, and ratings from Barron's on college selectivity. These were used to model another logistic regression model that could distinguish between a student who will attend a selective college and a student who will not attend a selective college 76% of the time when applied to data from new students. This model indicated that the following factors are associated with lower chance of attending a selective college: gaming the system, boredom, confusion, frustration, less engaged concentration, lower knowledge, and carelessness.

I finally looked at college major classification based on middle school student learning and engagement, specifically whether the major belonged to a STEM (Science, Technology, Engineering, Mathematics) or Non-STEM category. The logistic regression model developed could distinguish between a student who took a STEM college major and a student who took an non-STEM college major 66% of the time when applied to data from new students This model indicated that the following factors are associated with lower chance of enrolling in a STEM college major: gaming the system, lower knowledge, and carelessness.

4. Proposed Work

The initial individual models above support existing theories about indicators of successful entry to postsecondary education (academic achievement, grades). It sheds light on behavioral factors a student may experience in classrooms – which are more frequently and in many ways more actionable than the behaviors which result in disciplinary referrals – and how they can be predictive and be associated with long-term student outcomes.

With middle school assessments, I investigate at how student learning, academic emotions, and behavior as early as middle school may contribute as causal factors to a particular postsecondary decision (a in Figure 1 below) – an individual choice that is composed of answering the following questions: 1) Does the student decide to attend college?; 2) Does the student attend a selective college?; 3) What type of major does the student enroll in? I employ multivariate analysis on this part of my research work, for a richer and more realistic view of our postsecondary outcome, which is more than just one dependent variable of interest. Also this type of analysis allows us for causality to be deduced, as well as the inherent or underlying structure that can describe the data in a simpler fashion – in terms of latent variables. I also investigate interaction of features and how it affects our multivariate model via logistic regression, factor analysis and other appropriate statistical and machine learning algorithms that can be employed in our data to further understand the research problem.

In this phase of my dissertation research, I am starting to test the hypothesis of the possible existence of a mediating or indirect effect of high school college (and career) interests in predicting the multivariate postsecondary outcome based on middle school factors. I will establish this by looking at the causal influence of middle school factors to high school data (b in Figure 1 below). By integrating student data of their previous middle school interaction data, interests during their high school years, up to their postsecondary information, I will look at the possible causality of middle school factors to high school factors, as well as causality of high school factors to their postsecondary information. Like in previous analysis, I employ appropriate statistical and machine learning algorithms in trying to establish the indirect effect of high school factors (for our overall mediated

model later on). First, I look at how the middle school measures of student learning, engagement and academic emotions are predictive of the high school questionnaire responses, through multinomial logistic or decision tree algorithms. Then, I explore the association between the high school questionnaire responses with the multivariate postsecondary outcomes using structural equation modeling (factor analysis, regression, or path analysis).

Finally, by integrating emergent relationships and causal effects of middle school and high school factors on postsecondary outcomes conducted in the previous analyses, I will develop a multivariate predictive mediated model (c in Figure 1 below). Using student data that have complete information from middle school, to high school, to postsecondary years, I conduct causal modeling by fitting a mediational pathway model and evaluate how each of the variables influence one another over time [3]. In particular, using structural equation modeling (SEM), I develop a pathway starting from the middle school factors to the postsecondary outcomes, with high school factors as intervening or mediating factors. With significant zero-order correlations between the constructs (middle school factors, high school factors, postsecondary outcomes) established from the previous analyses, I employ a multiple regression analysis predicting postsecondary outcomes from both middle school and high school factors. It is expected that any partial effect (indirect effect) of high school factors (controlling for middle school factors) to be significant, decreasing the direct effect of middle school factors on postsecondary outcomes. Other SEM variants, such as factor analysis and path analysis are expected to be used as well for this analysis phase, to test the mediation model. This causal modeling has been used in educational research modeling motivational phenomena over time [3].

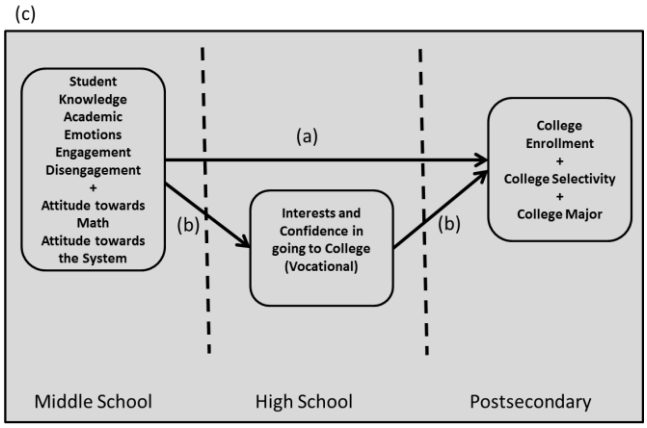


Figure 1. Modeling Postsecondary Outcomes from Middle School and High School factors: (a) Middle school factors predicting postsecondary outcomes; (b) Middle school factors predicting high school factors, High school factors predicting postsecondary outcomes; (c) Overall mediation model.

5. References

[1] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., and Koedinger, K. 2008. Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19, 2, 185-224.

[2] Bandura, A. 1997. *Self efficacy: The exercise of control*. New York,. NY: W. H. Freeman & Company.

[3] Blackwell, L., Trzesniewski, K., and Dweck, C. 2007. Implicit Theories of Intelligence Predict Achievement Across an Adolescent Transition: A Longitudinal Study and an Intervention. *Child Development*, 78, 1, 246-263.

[4] Borgen, F. and Betz, N. 2008. Career self-efficacy and personality: Linking the Career Confidence Inventory and the Healthy Personality Inventory. *Journal of Career Assessment*, 16, 22-43.

[5] Bowers, A. J. 2010. Grades and graduation: A longitudinal risk perspective to identify student dropouts. *The Journal of Educational Research*, 103, 3, 191-207.

[6] Chen, X. 2009. Students Who Study Science, Technology, Engineering, and Mathematics (STEM) in Postsecondary Education. Stats in Brief. *NCES 2009-161*. National Center for Education Statistics.

[7] College Division of Barron's Education Series (Ed.). 2012. *Barron's profiles of American colleges (30th ed.)*. Hauppauge, NY: Barron's Educational Series, Inc.

[8] Eccles, J. S., Vida, M. N., and Barber, B. 2004. The relation of early adolescents' college plans and both academic ability and task-value beliefs to subsequent college enrollment. *Journal of Early Adolescence*, 24, 63-77.

[9] Griffith, A. L. 2010. Persistence of women and minorities in STEM field majors: Is it the school that matters? *Economics of Education Review*, 29(6), 911-922.

[10] Lent, R. W., Brown, S. D., and Hackett, G. 1994. Toward a unifying social cognitive theory of career and academic interest, choice and performance. *Journal of Vocational Behavior*, 45, 1, 79–122.

[11] Lent, R. W., Lopez Jr, A. M., Lopez, F. G., and Sheu, H. B. 2008. Social cognitive career theory and the prediction of interests and choice goals in the computing disciplines. *Journal of Vocational Behavior*, 73, 1, 52-62.

[12] Mcquiggan, S. W., Mott, B. W., and Lester, J. C. 2008. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User modeling and user-adapted interaction*, 18, 81-123.

[13] Pekrun, R., Goetz, T., Daniels, L. M., Stupnisky, R. H., and Perry, R. P. 2010. Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology*, 102, 3, 531.

[14] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., and Rasmussen, K. P. 2005. The Assistment project: Blending assessment and assisting. In *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education*, 555-562.

[15] Roderick, M., Nagaoka, J., and Coca, V. 2009. College readiness for all: The challenge for urban high schools. *The Future of Children*, 19, 1, 185-210.

[16] Schmidt, W., Burroughs, N., Cogan, L., and Houang, R. 2011. Are College Rankings an Indicator of Quality Education? In *Forum on Public Policy Online*, 2011, 3. Oxford Round Table.

Who Do You Think I Am? Modeling Individual Differences for More Adaptive and Effective Instruction

Laura K. Allen
Arizona State University
Tempe, AZ, 85283
LauraKAllen@asu.edu

ABSTRACT
The purpose of intelligent tutoring systems is to provide students with personalized instruction and feedback. The focus of these systems typically rests in the adaptability of the feedback provided to students, which relies on automated assessments of performance in the system. A large focus of my previous work has been to determine how natural language processing (NLP)_ techniques can be used to model individual differences based on students' natural language input. My proposed research will build on this work by using NLP techniques to develop stealth assessments of students' individual differences and to provide more fine-grained information about the cognitive processes in which these students are engaged throughout the learning task. Ultimately, my aim will be to combine this linguistic data with on-line system data in order to develop more robust student models within ITSs for ill-defined domains.

Keywords
Intelligent Tutoring Systems, Natural Language Processing, Writing, Feedback, User Models

1. INTRODUCTION
The purpose of intelligent tutoring systems (ITSs) is to provide students with personalized instruction and feedback based on their performance, as well as other relevant individual characteristics [1]. The focus of these systems typically rests in the adaptability of the feedback provided to student users, which relies on automated assessments of students' performance in the system. Despite this adaptive feedback, however, many ITSs lack the ability to provide adaptive *instruction* and *higher-level feedback*, particularly when providing tutoring for ill-defined domains. This shortcoming is largely due to the increased difficulties associated with accurately and reliably assessing student characteristics and performance when the learning tasks are not "clear cut." In mathematics tutors, for instance, it can be relatively straightforward to determine when a student is struggling in specific areas; thus, these systems can provide adaptive instruction and feedback accordingly. For ITSs focused on ill-defined domains (such as writing and reading), on the other hand, this process can be more complicated. In particular, students' *open-ended* and *natural language* responses to these systems present unique assessment challenges. Rather than identifying a set of "correct" answers, the system must identify and analyze characteristics related to students' responses in order to determine the quality of their performance as well as the areas in which they are struggling.

Natural language processing (NLP) techniques have been proposed as a means to target this assessment problem in adaptive systems. In particular, NLP provides detailed information about the characteristics of students' natural language responses within these systems [2] and subsequently helps to model students' particular areas of strengths and weaknesses [3]. NLP has begun to be incorporated within ITSs more frequently [4-5] because it allows systems to automatically evaluate the quality and content of students' responses [6-7]. Additionally, these assessments afford systems the opportunity to model students' learning throughout training and subsequently improve models of their performance [8]. Previous research suggests that these NLP techniques can increase the efficacy of computer-based learning systems. In particular, NLP helps to promote greater interactivity in the system and, consequently, leads to increased learning gains when compared to non-interactive training tasks (e.g., reading books, watching videos, listening to lectures [5, 9].

In my previous research, my colleagues and I have proposed that NLP techniques can be used to determine much more than simply the *quality* of a particular response in the system. Specifically, NLP can serve as a powerful methodology for modeling individual differences among students, as well as for examining the specific processes in which these students are engaging [3, 8]. In this overview, I suggest that, when combined with *on-line* interaction data, these NLP techniques can provide critical information that can be used to enhance the adaptability of ITSs, particular those focused on ill-defined domains. Thus, the aim of my research is to investigate how the linguistic characteristics of students' language can provide a window into their cognitive and affective processes. This information will then be combined with system data to promote more personalized learning experiences for the student users in these systems.

1.1 Writing Pal
The Writing Pal (W-Pal) is a tutoring system that was designed for the purpose of increasing students' writing proficiency through explicit strategy instruction, deliberate practice, and automated feedback [10]. In the W-Pal system, students are provided explicit

strategy instruction and deliberate practice throughout eight instructional modules, which contain strategy lesson videos and educational mini-games. The instruction in these modules covers specific topics in the three main phases of the writing process—prewriting (*Freewriting*, *Planning*), drafting (*Introduction Building*, *Body Building*, *Conclusion Building*), and revising (*Paraphrasing*, *Cohesion Building*, *Revising*).

Animated pedagogical agents narrate the W-Pal lesson videos by providing explicit descriptions of the strategies and examples of how these strategies can be used while writing (see Figure 1 for screenshots). The content covered in these videos can be practiced in one or more of the mini-games contained within each module. The purpose of these mini-games is to offer students the opportunity to practice the individual writing strategies without having to compose an entire essay.

Figure 1. Screenshots of the W-Pal Lesson Videos

W-Pal contains an AWE component in addition to the eight instructional modules, where students can practice holistic essay writing. This component of W-Pal contains a word processor where students can compose essays and automatically receive summative (i.e., holistic scores) and formative (i.e., actionable, strategy-based) feedback on these essays. The *summative* feedback in W-Pal is calculated using the *W-Pal assessment algorithm*. This algorithm employs linguistic indices from multiple NLP tools to assign essays a score from 1 to 6 (for more information, see 11). The purpose of the *formative* feedback is to teach students about high-quality writing and to provide them with actionable strategies for improving their essays. To deliver this feedback, W-Pal first identifies weaknesses in students' essays (e.g., essays are too short; essays are unorganized). It then provides students with feedback messages that designate specific strategies that can help them to work on the problems. Previous studies have demonstrated that W-Pal is effective at promoting increases in students' essay scores over the course of multiple training sessions [6; 12].

1.2 Current Work

The focus of my doctoral research will be on the use of NLP techniques to develop stealth assessments of students' individual differences and to provide more fine-grained information about the cognitive processes in which these students are engaged throughout the learning task. Ultimately, the aim of this research will be to combine this linguistic data with on-line system data in order to develop more robust student models within ITSs for ill-defined domains, such as W-Pal.

The goal of this specific research project will be to use the linguistic properties of students' essays to model individual differences related to writing performance (e.g., vocabulary knowledge). This data will then be combined with *on-line* process data, such as students' keystrokes while writing, to provide a more complete understanding of their writing processes. Ultimately, this project will aim to determine whether there are specific writing processes (as identified by the *characteristics* of the essays and students' *on-line processes*) that are more or less predictive of successful writing and revision. My final goal will then be to use this information to provide more adaptable instruction and formative feedback to students.

2. Proposed Contributions of Current Work

This proposed research project will contribute to both the W-Pal system, as well as the EDM community more generally. Regarding the W-Pal system, the development of stealth assessments and online student models will significantly enhance the adaptability and, theoretically, the efficacy of the system. The current version of W-Pal does not provide individualized instruction to students and only adapts the feedback based on single (i.e., isolated) essays that they generate. Thus, the system does not consider students' previous interactions with the system when providing feedback, nor the individual characteristics of these student users. Therefore, the proposed work will help to provide a much more robust student model, which should help W-Pal provide more personalized instruction and feedback.

More generally, the results of this project (and future projects) will contribute to the EDM community, as well as to research with natural language data more broadly. Language is pervasive and, here, we propose that it can be used to provide *unique* information about individuals' behaviors, cognitive processes, and affect. By investigating the specific characteristics of students' natural language data, we can glean important insights about their learning processes, beyond information that can be extracted from system log data. By combining NLP with other forms of data, researchers will gain a more complete picture of the students using the system, which should ultimately lead to more effective instruction.

3. Previous Work

A large focus of my previous work has been to determine how NLP techniques can be used to model individual differences based on students' natural language input. Importantly, this input has ranged from more structured language (such as essays) to naturalistic language responses (such as self-explanations). As an example, in one study, my colleagues and I investigated whether we could leverage NLP tools to develop models of students' comprehension ability based on the linguistic properties of their self-explanations [3]. Students (n = 126) interacted with a reading comprehension tutor where they self-explained target sentences from science texts. Coh-Metrix [13] was then used to calculate the linguistic properties of these aggregated self-explanations. The results of this study indicated that the linguistic indices were predictive of students' reading comprehension ability, over and above the current system algorithms (i.e., the self-explanation scores). These results are important, because they suggest that NLP techniques can inform stealth assessments and help to improve student models within ITSs.

In further research projects, we have begun to investigate how these linguistic characteristics change across time, and how these changes relate to individual differences among the students [14].

In particular, we proposed that the *flexibility* of students' writing style could provide important information about their writing proficiency. In one study, we investigated college students' (n = 45) flexibility in their use of cohesion across 16 essays and whether this flexibility related to their writing proficiency. The results suggested that more proficient writers were, indeed, more flexible in their use of cohesion across different writing prompts and that this cohesive flexibility was most strongly related to the unity, or coherence, of students' writing. The results of this study indicated that students might differentially employ specific linguistic devices in different situations in order to achieve coherence among their ideas. Overall, the results of these (and many other) studies provide preliminary evidence that NLP techniques can be used to provide unique information about students' individual differences and learning processes within ITSs.

4. Advice Sought

I am seeking advice for my proposed research regarding two primary questions. First, *what analytical methods should be used to most effectively model individual differences based on linguistic data?* In previous research, my colleagues and I have relied heavily on stepwise regression and discriminant function analysis techniques to model students' essay scores and individual differences. However, this technique can pose particular problems and is not always the most effective regarding large-scale data sets containing many variables, such as these. Thus, I would largely benefit from expert advice regarding the specific modeling techniques that can help to improve this research.

My second question relates to: *what on-line process data can be most effectively tied with this linguistic data – and how?* In previous studies, we have heavily relied on the linguistic properties of students' responses alone to model and understand the learning process. However, these models could be greatly strengthened through the addition of on-line processing data, such as keystrokes or eye tracking. We have begun to implement keystroke logging into the W-Pal system to begin to investigate this question. However, I would greatly benefit from expert advice regarding the best methods for combining this data into a reliable and accurate student model.

5. ACKNOWLEDGMENTS

Prior research was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080589 to Arizona State University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. I would also like to thank Danielle McNamara, Scott Crossley, Erica Snow, Jennifer Weston, Rod Roscoe, and Matthew Jacovina for their contributions to this line of work.

6. REFERENCES

[1] Murray, T. 1999. Authoring intelligent tutoring systems: An analysis of the state of the art. International *Journal of Artificial Intelligence in Education,* 10, (1999) 98-129.

[2] Crossley, S. A., Allen, L. K., and McNamara, D. S. 2014. Analyzing discourse processing using a simple natural language processing tool (SiNLP). *Discourse Processes,* 51, (2014) 511-534.

[3] Allen, L. K., Snow, E. L., and McNamara, D. S. in press. Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques. *Proceedings of the 5th International Learning Analytics and Knowledge Conference*.

[4] Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A. and Louwerse, M. 2004. AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, 36*, (2004) 180-193.

[5] VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., and Rose, C. P. 2007. When are tutorial dialogues more effective than training? *Cognitive Science, 31*, (2007), 3-62.

[6] Crossley, S. A., Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Using automated indices of cohesion to evaluate an intelligent tutoring system and an automated writing evaluation system. In K. Yacef et al (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED).* Springer, Heidelberg, Berlin, 269-278.

[7] Rus, V., McCarthy, P., Graesser, A. C., and McNamara, D. S. 2009. Identification of sentence-to-sentence relations using a textual entailer. *Research on Language and Computation, 7,* (2009) 209-229.

[8] Varner, L. K., Jackson, G. T., Snow, E. L., and McNamara, D. S. 2013. Are you committed? Investigating interactions among reading commitment, natural language input, and students' learning outcomes. In S. K. D'Mello, R. A., Calvo, & A. Olney (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining*. Springer, Heidelberg, Berlin, 368-369.

[9] Graesser, A. C., McNamara, D. S., and Rus, V. 2007. Computational modeling of discourse and conversation. In M. Spivey, M. Joanisse, & K. McRae (Eds.), *Cambridge Handbook of Psycholinguistics*. Cambridge University Press, Cambridge, UK.

[10] Roscoe, R. D., Allen, L. K., Weston, J. L., Crossley, S. A., and McNamara, D. S. 2014. The Writing Pal intelligent tutoring system: Usability testing and development. *Computers and Composition*, 34, (2014) 39-59.

[11] McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., and Dai, J. 2015. Hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23, (2015), 35-59.

[12] Allen, L. K., Crossley, S. A., Snow, E. L., and McNamara, D. S. 2014. Game-based writing strategy tutoring for second language learners: Game enjoyment as a key to engagement. *Language Learning and Technology*, 18 (2014), 124-150.

[13] McNamara, D. S., Graesser, A. C., McCarthy, P., and Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix.* Cambridge: Cambridge University Press.

[14] Allen, L. K., Snow, E.L., and McNamara, D. S. 2014. The long and winding road: Investigating the differential writing patterns of high and low skilled writers. In J. Stamper, S. Pardos, M. Mavrikis, & B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (pp. 304-307). London, UK.

Developing Self-Regulated Learners Through an Intelligent Tutoring System

Kim Kelly
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
508-461-6386
kkelly@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
508-831-5569
nth@wpi.edu

1. ABSTRACT

Intelligent tutoring systems have been developed to help students learn independently. However, students who are poor self-regulated learners often struggle to use these systems because they lack the skills necessary to learn independently. The field of psychology has extensively studied self-regulated learning and can provide strategies to improve learning, however few of these include the use of technology. The present proposal reviews three elements of self-regulated learning (motivational beliefs, help-seeking behavior, and meta-cognitive self-monitoring) that are essential to intelligent tutoring systems. Future research is suggested, which address each element in order to develop self-regulated learning strategies in students while they are engaged in learning mathematics within an intelligent tutoring system.

2. KEYWORDS

Intelligent tutoring systems, self-regulated learning, meta-cognition

3. DEFINING THE PROBLEM

Intelligent tutoring systems (ITS) are designed to provide independent learning opportunities for students. Learning occurs through hints, tutoring, scaffolding and correctness feedback. A great body of research exists surrounding types and timing of feedback [6] and tutoring that have been found to improve student outcomes. As a classroom teacher I have used several different ITS with students to help them learn mathematics. Over the years, I have seen many students benefit from these systems. However, I have also witnessed students struggling to use the systems and who fail to learn, despite all of the assistance provided. Addressing this failure serves as the basis of my dissertation. For an ITS to achieve maximum results, the students using the system must be good self-regulated learners. My proposed research attempts to use an ITS to develop self-regulating strategies, while students are learning the desired content.

Zimmerman and Campillo [12], suggest that self-regulated learning is a three-phase process. During the *Forethought Phase*, students engage in a task analysis, which includes goal setting and strategic planning. Self-motivational beliefs, including self-efficacy [11, 4] outcome expectations, task value/interest [10], and goal orientation also play a significant role in this phase as they have been found to positively affect student learning. During the *Performance Phase*, students demonstrate self-control by employing various task strategies and help-seeking behaviors. Self-observation, which includes meta-cognitive self-monitoring, is also crucial. During the final phase, *Self-Reflection*, students engage in self-judgment and self-reaction.

2. PROPOSED SOLUTION

To help develop self-regulated learners, these components must be explicitly taught. However, some aspects are seemingly more relevant than others when interacting with an ITS. Specifically motivational beliefs, help-seeking behavior, and meta-cognitive self-monitoring can all be addressed within the structures of intelligent tutoring systems. The following sections discuss each of these components by presenting relevant literature, sharing results of my previously published studies, and proposing future research components of my dissertation.

2.1 Motivational Beliefs

One aspect of the first phase of self-regulated learning is motivation. Students who are strong self-regulated learners have high self-efficacy. Schunk [11] defines self-efficacy as "an individual's judgment of his or her capabilities to perform given actions." A student's belief that they are capable of learning can be influenced by a growth mindset [4]. Some of my earlier research, using teacher-created motivational videos, attempted to create a growth mindset in students while they were completing math homework inside of an intelligent tutoring system [7]. While the minimal intervention failed to show changes in student self-reports of mindset, there was a significant increase in the perception of task value and homework completion rates as a result of a video inspired by [10]. In addition to improving self-efficacy, increasing task value/interest is important to developing self-regulated learners. The protocol employed in my initial study is promising and a more sophisticated intervention will be explored to further increase motivation.

2.2 Help Seeking Behaviors

Intelligent tutoring systems provide many different structures to support student learning. One such structure that I have explored is correctness-only feedback. I found that this simple support provided by an ITS during a homework assignment was found to improve student learning significantly compared to traditional paper and pencil homework that did not provide immediate feedback [8]. Yet research has shown that many students do not effectively take advantage of these features. Aleven et al. [1] explores ineffective help use in interactive learning environments and suggests that there are system-related factors, student-related factors and interactions between these factors that impact help-seeking behaviors. In one of my recent studies, I found that there

are students who, despite access to the same instructional supports, do not successfully take advantage of them and therefore do not learn [9]. This has resulted in a phenomenon called wheel spinning [3], where students persist without making progress towards learning. I hypothesize that wheel spinning is a result of ineffective help-seeking behaviors. Therefore, I propose a study that would provide direct interventions to teach students the necessary help-seeking behaviors to become self-regulated learners.

2.3 Meta-Cognitive Self-Monitoring

Elements of meta-cognition, are evident in all three phases of self-regulated learning. For example, goal setting is prominent in phase one. Other elements, like self-monitoring, are evident in multiple phases. Self-monitoring involves students becoming aware of their performance and judging their knowledge. This is sometimes referred to as metacognitive knowledge monitoring [5]. In phase two, while students are participating in a learning task, they must monitor what they are learning. Students who are strong self-regulated learners will seek feedback to easily monitor their progress. I surveyed my students to better understand their perception of feedback. High performing students claimed that the immediate feedback provided by an ITS caused frustration, but was also beneficial to their learning [8]. They were able to identify their mistakes and learn from them. To help all students recognize the importance of monitoring their learning, I propose a study where students are provided feedback along with progress monitoring to show the benefits.

Self-monitoring continues into the third phase of self-regulated learning. During this reflection stage, students assess their success or failure. Strong self-regulated learners may challenge themselves in some way to confirm their success. A willingness to seek out challenges ties back into the growth mindset that is addressed in phase one. Students who believe that intelligence is fixed will often shy away from challenges for fear of failure, whereas students with a growth mindset view challenges as opportunities to learn more [4]. Therefore, to encourage all students to seek out challenges as a method to self-monitor, I propose a study where growth mindset messages are embedded in ITS and opportunities for students to choose challenging problems are provided.

3. CONTRIBUTION

Intelligent tutoring systems rely on independent learning practices to effectively teach students. For example, students must use available hints and tutoring to navigate new material. However not all students successfully learn when using an ITS. Some early research suggests that these students are those who struggle with self-regulated learning. The field of psychology has studied self-regulated learning for more than a decade, resulting in many ideas that can improve instruction. Some ITS have incorporated features to help students who lack self-regulated learning strategies, like automatically detecting when a student is frustrated [2] and providing additional assistance when a student is failing. However, little research has explored how technology can actually promote self-regulated learning. By integrating the capabilities of intelligent tutoring systems with the vast knowledge of self-regulated learning, the proposed research seeks to teach students how learn effectively. By addressing specific aspects of self-regulated learning, ITS can actually teach students how to learn while teaching them content.

This paper is part of my dissertation proposal and is being submitted as a doctoral consortium paper to the Artificial Intelligence In Education Conference (2015) and the Educational Data Mining Conference (2015).

4. REFERENCES

[1] Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. (2003). Help Seeking and Help Design in Interactive Learning Environments. *Review of Educational Research*, 73(3), 277-320.

[2] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., Koedinger, K. (2008) Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. *Journal of Interactive Learning Research*, 19 (2), 185-224

[3] Beck, J., & Gong, Y. (2013). Wheel-Spinning: Students Who Fail to Master a Skill. In *Proceedings of the 16th Artificial Intelligence in Education*, Lane, H.C., Yacef, K., Mostow, J., & Pavlik, P. (Eds.) 431-440.

[4] Dweck, C. (2006). *Mindset*. New York: Random House.

[5] Isaacson, R. M., & Fujita, F. (2006). Metacognitive Knowledge Monitoring and Self-Regulated Learning: Academic Success and Reflections on Learning. *Journal of the Scholarship of Teaching and Learning*, 6(1), 39-55.

[6] Kehrer, P., Kelly, K. & Heffernan, N. (2013). Does Immediate Feedback While Doing Homework Improve Learning. In Boonthum-Denecke, Youngblood(Eds) *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press 2013. p 542-545.

[7] Kelly, K., Heffernan, N., D'Mello, S., *Namias*, J., & Strain, A. (2013a). Adding Teacher-Created Motivational Video to an ITS. *Florida Artificial Intelligence Research Society* (FLAIRS 2013). 503-508.

[8] Kelly, K.. Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, J., Soffer Goldstein, D., (2013b). Estimating the Effect of Web-Based Homework. In Lane, Yacef, Motow & Pavlik (Eds) *The Artificial Intelligence in Education Conference*. Springer-Verlag. pp. 824-827.

[9] Kelly, K., Wang, Y., Thompson, T., Heffernan, N. (2015). Defining Mastery: Knowledge Tracing Versus N-Consecutive

Correct Responses. Submitted to Educational Data Mining Conference: Madrid Spain (2015).

[10] Pekrun, R. (2006). The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Educational Psychology Review, 18*(4), 315-341.

[11] Schunk, D. H. (1996). Self-efficacy and academic motivation. *Educational Psychologist*, 26, 207-231.

[12] Zimmerman, B. J. & Campillo, M. (2002). Motivating self-regulated problem solvers. In J. E. Davidson & R. J. Sternberg (Eds.), *The nature of problem solving*. New York: Cambridge University Press.

Data-driven Hint Generation from Peer Debugging Solutions

Zhongxiu Liu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
zliu24@ncsu.edu

ABSTRACT

Data-driven methods have been a successful approach to generating hints for programming problems. However, the majority of previous studies are focused on procedural hints that aim at moving students to the next closest state to the solution. In this paper, I propose a data-driven method to generate remedy hints for BOTS, a game that teaches programming through a block-moving puzzle. Remedy hints aim to help students out of dead-end states, which are states in the problem from where no student has ever derived a solution. To address this, my proposed work includes designing debugging activities and generating remedy hints from students' solutions to debugging activities.

1. INTRODUCTION

Programming problems are characterized by huge and expanding solution spaces, which cannot be covered by manually designed hints. Previous studies have shown fruitful results in applying data-driven approaches to generate hints for programming problems. Barnes and Stamper [1] designed the Hint Factory, which gives student feedback using previous students' data. The Hint Factory uses a data structure called an interaction network as defined by Eagle et al. [3], in which nodes represent the program states and edges represent the transitions between states. Peddycord et al. [7] applied the Hint Factory in BOTS, a game that teaches programing through block-moving puzzles. This study introduced worldstates, which represent the output of a program, and compared them to codestates, snapshots of the source code. This study found that using interaction networks of worldstates can generate hints for 80% of programming states. Rivers and Koedinger [9] applied the Hint Factory in a solution space where snapshots of students' code (program state) are represented as trees, and trees are matched when the programs they represent are within a threshold of similarity. Piech et al. [8] applied data-driven approach to programs from a MOOC. This work compared the methods in Rivers and Koedinger's[9] and Barnes's[1] studies, together with algorithms that predict the desirable moving direction from a program state and generate hints to push students toward the desirable direction.

However, previous studies mainly focused on generating procedural hints that direct students to the next program state. Data from previous students' work may be insufficient to provide a next-step hint from a "dead-end state". Second, even if a next-step hint could be generated, simply telling students where to move next is not enough. An example of this situation is shown in Figure 2 - if a student follows a path that leads to a dead-end state (marked in blue), then the only hint we are able to offer is to delete all work since the last branching point. This may be a bad advice; just because we have not seen a student solve the problem this way does not mean that the solution is incorrect. Even with a correct solution down this path, we are unlikely to see it since most students solved the problem in a more conventional way, either because they have a better understanding of the problem or because our hints guide them towards the more conventional solution. Thus, students in dead-end states, who may actually have a correct solution in mind, are unable to receive helpful hints.

In this paper, I propose a data-driven method to generate remedy hints in Bots. Remedy hints are hints that help students in dead-end states by telling them why their current state is wrong, and where to move from their current state. To address the problem of insufficient data, I will collect data from debugging activities in BOTS, where students work out solutions from dead-end states and provide explanations. I hypothesize that this study will not only help students who are wheel-spinning on dead-end states, but also the students who are providing debugging solutions.

2. RESEARCH METHODOLOGY

2.1 Designing Debugging Activities

Debugging activities will be designed as bonus challenges for students who successfully complete a level. The content of debugging activities will be the dead-end states from the problem they completed. Given a dead-end state, a student will first be asked to explain the error in the program, and why it led to the dead-end state. The student will then be asked to explain his/her debugging strategy. Lastly, the student will apply his/her debugging strategy and fix the program from its current state to a goal state. In this process, both the student-written explanations and the transitions of program states will be used as hints. A more detailed explanation of these are explained in the following section.

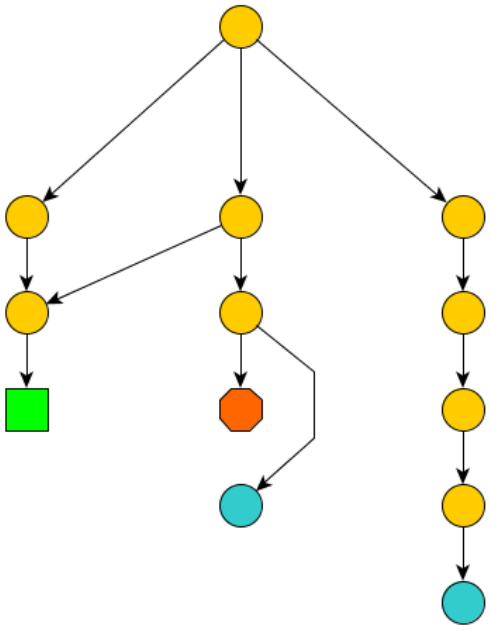


Figure 1: Interaction Network in BOTS.Green is the solution state;orange is an error state (e.g. the robot runs off the stage);blue are the dead-end states; yellow represents the rest states

To encourage students to participate in debugging activities, I will introduce a voting system. Completing debugging activities will earn points or advantages from the game. Currently, BOTS applies a rewarding system for students who solve the puzzle with fewer lines of code, as shown in Figure 2. On the left is the optimal number of lines of code needed to solve the puzzle. On the right is the current player's record for the fewest lines of code. Players earn 4 stars for reaching the optimal solution, 3 stars for being within a certain threshold value, down to one star for merely completing the puzzle. Additionally, clicking the optimal solution shows the name of the first user to reach the optimal solution.

I will design a similar leaderboard to reward students who used fewer steps when debugging for a dead-end state. Encouraging students to use fewer steps will reduce the size of debugging solutions, and the likelihood that a student will delete previous work and start from scratch. Moreover, students will receive rewards for writing good quality explanations on states and debugging strategies. The quality will be measured by a voting mechanism. Students who received a student-written explanation will be able to vote for the hint as "helpful." The more votes an explanation receives, the more points its author will get. Students with the most points will have their names appear in a leaderboard.

2.2 Construct Hint from Debugging Work

Completing a debugging problem is defined as successfully moving from the current state to the final goal state. The debugging process will be treated as a self-contained problem with its own local interaction network. When completed, this local interaction network will be added to the global interaction network for the problem. With a more complete

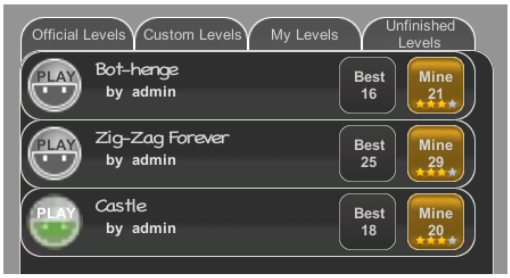


Figure 2: BOTS rewarding system

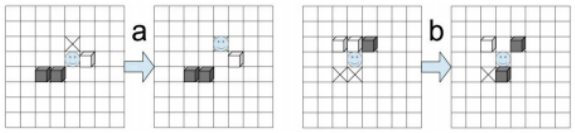


Figure 3: Two generated hints for a simple puzzle. The blue is the robot. The 'X' is a goal. Shaded boxes are boxes placed on goal spot. Not shaded boxes are not on goal spot.

global interaction network, Hint Factory [1] can be applied to generate hints for previously dead-end states.

Student-written explanations will be presented together with hints generated by the Hint Factory. An example hint from the current BOTS system is shown in Fig 3. Before presenting the hints from the Hint Factory, a student in dead-end state will see a student-written explanation on where and why their current program is wrong. This will give students a chance to reflect on their own program. Then, the student can request to see a student-written explanation of the debugging plan for the current state. This will enable the student to solve the problem on their own following a debugging plan, instead of blindly following procedural hints.

When multiple debugging approaches are available for a state, I will experiment with selecting the best debugging solution to generate hints. Ideally, I would select a debugging approach with the shortest solution path. However, there may be situations where students debugged by starting over from the beginning, which may or may not be the best solution. One approach is to evaluate the path that leads toward the current state. Assume there is a failure state in the student's solution; the earlier this failure state occurs in the path, the more likely the solution is wrong from the start and back-to-start is a good solution.

When multiple student-written explanations are available for a debugging solution, I will start by randomly choosing one explanation. As the voting process goes, I will filter out the explanations with significantly lower 'helpful' votes.

3. EVALUATION

My evaluation will focus on the below research questions:

- What percentage of students will participate in the debugging activities, and how many write explanations? Why do students participate or not?

- What is the relationship between students' involvement in debugging and their programming performance? Will students who complete problems with shorter solutions be more involved in debugging?

- Will writing or reading student-written explanations and debugging strategies help learning?

- In the global interaction network, what percentage of the dead-end program states receive hints from student debugging solutions?

Previous BOTS participants are students from after-school programming education activities. In my experiment, I will randomly recruit the same type of students. These students will be separated into a control group where students will use the traditional BOTS system, an experimental group A where students will be given the option to do debugging challenges, and an experimental group B where students must do debugging challenges after completing a level.

To answer the first research question, students from the two experimental groups will do a post survey on their opinions about debugging activities and hints generated from student-written explanations. For experimental group A, I will add survey questions on why students chose to participate or not participate in debugging activities. To answer the second question, students' interaction and compilation data while playing BOTS will be recorded. These data will be used to measure the relationship between involvement in the debugging activities and programming performance. To answer the third research question, students from all groups will do pre and post-tests on basic programming and debugging concepts that are related to BOTS content. Learning gains will be measured as the difference between pre and post-test. To answer the fourth question, the program state space coverage will be compared between the three groups.

4. PROPOSED CONTRIBUTION

My work will generate a new type of hint that may lead to different pedagogical results than the procedural hint, especially for students in dead-end states. My work will demonstrate the feasibility of collecting data from peer students' debugging processes, and generating helpful hints.

My work will design a feature that supports both programming and debugging activities in an educational game. This design will have several pedagogical benefits. First, Kinnunen and Simon's[6] research have shown that novice programmers experienced a range of negative emotions after errors. Practicing debugging will help novice programmers proceed after errors, and enjoy programming experiences. Second, students will make self-explanations on the observed flaw and debugging strategy, and decades of research such as Johnson and Mayer's[5], and Chi et al.[2] have shown that self-explanation is extremely beneficial to learning. Third, students in dead-end states will not only receive help, but also learn what peer students think given the same situation.

5. ADVICE SOUGHT

Johnson and Mayers[5], and Hsu et al. studies[4] have shown that merely adding self-explanation features did not help learning, but students' engagement in self-explaining did. Therefore, I want to seek advice on the design of debugging activities that engage students in debugging and writing explanations, and produce quality work. I also want to seek advice on the evaluation. Given the previous question, how should I measure the level of engagement in debugging and self-explaining?

Moreover, introducing debugging challenges as extra activities will affect other measurements. For example, students who spend a significant amount of time in debugging may complete less problems given the time constraint, and exhaust earlier. How should I address this problem and measure students' performance fairly? Moreover, how to design pre and post-tests to measure learning gains from debugging process? Lastly, what are the potentials, benefits, and risks to expand this work into programming problems using mainstream programming languages?

6. REFERENCES

[1] T. Barnes and S. John. Toward automatic hint generation for logic proof tutoring using historical student data. In *Proceedings of the 6th International Conference on Intelligent Tutoring System*, pages 373–382, 2008.

[2] M. T. Chi, N. Leeuw, M. H. Chiu, and C. LaVancher. Eliciting self–explanations improves understanding. *Cognitive Science*, 18(3):439–477, 1994.

[3] M. Eagle, M. Johnson, and T. Barnes. Interaction networks: Generating high level hints based on network community clusterings. In *Proceedings of the 6th International Conference on Intelligent Tutoring System*, pages 164–167, 2012.

[4] C. Y. Hsu, C. C. Tsai, and H. Y. Wang. Facilitating third graders' acquisition of scientific concepts through digital game–based learning: The effects of self–explanation principles. *The Asia–Pacific Education Researcher*, 21(1):71–82, 2012.

[5] C. I. Johnson and R. E. Mayer. Applying the self-explanation principle to multimedia learning in a computer-based game–like environment. *Computers in Human Behavior*, 26(6):1246–1252, 2010.

[6] P. Kinnunen and B. Simon. Experiencing programming assignments in cs1: the emotional toll. In *Proceedings of the 6th international workshop on Computing education research*, pages 77–86, 2010.

[7] B. Peddycord III, A. Hicks, and T. Barnes. Generating hints for programming problems using intermediate output. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 92–98, 2014.

[8] C. Piech, M. Sahami, J. Huang, and L. Guibas. Autonomously generating hints by inferring problem solving policies. In *Proceedings of Learning at Scale*, 2015.

[9] K. Rivers and K. R. Koedinger. Automatic generation of programming feedback: A data-driven approach. In *Proceedings of the 1st workshop on AI-supported Education for Computer Science, th 16th International Conference on Artificial Intelligence on Education*, pages 50–59, 2013.

Enhancing Student Motivation and Learning Within Adaptive Tutors

Korinn S. Ostrow
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
ksostrow@wpi.edu

ABSTRACT

My research is rooted in improving K-12 educational practice using motivational facets made possible through adaptive tutoring systems. In an attempt to isolate best practices within the science of learning, I conduct randomized controlled trials within ASSISTments, an online adaptive tutoring system that provides assistance and assessment to students around the world. My work also incorporates big data analytics through the establishment of data driven learning models that promote the use of finite assessment to optimize student modeling and enhance user motivation. This paper highlights a turning point in my research as I transition into PhD candidacy. My contributions thus far and my research goals are discussed, with consult sought on how to best meld the realms of my work moving forward. An iteration of this work has also been published as a Doctoral Consortium at AIED 2015 [4].

Keywords

Motivation, Learning, Feedback, Choice, Assessment Methodologies, Adaptive Tutoring

1. RESEARCH FOCUS

1.1 Adaptive Tutoring: ASSISTments

The U.S. Department of Education's National Educational Technology Plan supported the idea that technology will play a key role in delivering personalized educational interventions [14]. Yet there remains a severe lack of research regarding the effectiveness of online learning systems for K-12 education [15]. Adaptive tutoring systems offer interactive learning environments that allow students to excel while providing teachers a unique approach to classroom organization and data-driven lesson plans. Before the development of these adaptive platforms, research within classrooms was costly and generally required a longitudinal approach. As such, much of the evidence that supports K-12 educational practice is generalized from studies conducted by psychologists in laboratory settings with college undergraduates.

My research acts on this deficit, by conducting controlled trials using student level randomization within ASSISTments, an online adaptive tutoring system, to isolate best practices for learning outcomes while enriching the user experience. ASSISTments, commonly used for both classwork and homework, presents students with immediate feedback and a variety of rich tutorial strategies. The platform is also a powerful assessment tool, providing teachers with a variety of student and class reports that pinpoint where students are struggling and enhance classroom techniques using real time data. Further, the platform is unique in that it allows educational researchers to design and implement content-based experiments without extensive understanding of computer programming, serving as a shared collaborative tool for the advancement of the science of learning [3].

1.2 Motivational Trinity

Essentially, my work seeks to enhance student motivation and performance by enriching content through optimized feedback delivery, exploring opportunities to make students shareholders in the learning process, and attempting to boost motivation and proper system usage through improved assessment techniques.

1.2.1 Feedback Mediums

Until recently, virtually all feedback within the ASSISTments tutoring platform was provided using text, typically with font color or typeset signifying important variables. However, adaptive tutoring systems offer the opportunity to utilize a variety of hypermedia elements, as outlined by Mayer's multimedia principles for the optimal design of e-Learning environments [1]. These twelve principles, driven by cognitive theory, promote active learning while reducing cognitive load and accounting for the average user's working memory [1]. Educational technologies that employ video tend to do so in a manner that resembles lectures rather than feedback (i.e., Khan Academy). Thus, the introduction of matched content video feedback to the ASSISTments platform through brief 15-30 second YouTube recordings offered a novel approach to investigating hypermedia within an adaptive setting.

1.2.2 Student Choice

While platforms like ASSISTments offer a variety of features, few make students shareholders in the learning process. Despite the fact that users can endlessly customize their experiences with commercial products, student preference is not a key element in the realm of education. Choice is an intrinsically motivating force [11] that has the potential to boost subjective control, or a student's perception of their causal influence over their learning outcomes [12]. Feelings of control are balanced by appraisals of subjective value, or a student's perceived importance of her learning outcome. By providing the student with choices at the start of her assignment, it may be possible to enhance

expectancies regarding her performance and thereby enhance achievement emotions such as motivation [12]. Considering the control-value theory within the realm of an adaptive tutoring system for mathematics content may help to explain and ameliorate female dropout in STEM fields [2]. Feedback medium personalization offers one simple method to examine the motivational effect of choice within these platforms.

1.2.3 Improving Assessment

Adaptive tutoring systems typically function through measures of binary correctness on a student's first attempt or first action within a problem. Within such systems, students who take advantage of tutoring feedback are unduly penalized. This creates an environment in which students are afraid to use the beneficial features of these platforms, or instead, overuse feedback if they have already lost credit (i.e., skipping to the answer rather than reading a series of hints). The establishment of partial credit scoring would help to alleviate these issues, serving to motivate student performance while simultaneously offering teachers a more robust view of student knowledge. Using data mining approaches, partial credit can be defined algorithmically [16] for the purpose of enhancing student modeling. Real time implementation of these data driven models could offer substantial benefits for all parties.

2. PROPOSED CONTRIBUTIONS

Thus far, my work has lead to eight peer reviewed articles already published or in press, as well as a multitude of projects that are in progress. Projects that best highlight my goals as I transition to my PhD work are described in the following subsections.

2.1 Published Works

2.1.1 Video vs. Text Feedback

The ASSISTments platform was used to conduct a randomized controlled trial featuring matched content video and text feedback within the realm of middle school mathematics [7]. Results suggested significant effects of video feedback, showing enhanced learning outcomes on next question performance after receiving adaptive video tutoring, as well as increased efficiency. Further, through self-report it was observed that students perceived video as a positive addition to their assignment. This study was the first of its kind to explore the potential for replacing text feedback, already shown to be successful within ASSISTments [13], with an alternate medium. A scaled-up replication of this study is currently underway. This work inspired an influx of video content into the ASSISTments platform, providing new opportunities to examine the subtleties of video feedback, including a crowd-sourced approach to feedback creation.

2.1.2 Dweckian Motivation

Moving beyond the use of video feedback and into the realm of pedagogical agents, my co-authors and I sought to investigate the motivational effects of Dweckian inspired mindset training within ASSISTments feedback [10]. A six-condition design was used to examine how growth mindset messages promoting the malleability of intelligence delivered with domain based feedback effected motivation and learning outcomes. Conditions differed on elements of audiovisual message delivery, ranging from plain text to an animated pedagogical agent. Although limited by a small sample size and ceiling effects, analyses across five mathematics skills revealed that mindset messages altered student performance as measured by persistence, learning gain, and self-reported enjoyment of the system (trends, $p \approx 0.1$). Trends also pinpointed gender differences in response to messages delivered using the pedagogical agent.

2.1.3 Partial Credit Assessment

By data mining log files from ASSISTments usage spanning the 2012-2013 school year, this work established a simple student modeling technique for the prediction of next problem correctness (time $t + 1$) using algorithmically defined partial credit scores at time t [5]. Although traditional modeling approaches and most adaptive tutors are driven by binary metrics of student correctness, employing partial credit can enhance student motivation and promote proper use of system features such as adaptive feedback, while allowing teachers a more robust understanding of student ability and simultaneously enhancing predictive modeling. Predictions gathered using a tabling approach based on maximum likelihood probabilities were able to compete with standard Knowledge Tracing models in terms of model accuracy, while drastically reducing computational costs [5].

2.2 Works in Press or in Progress

2.2.1 Student Choice

This work served as a pilot study on the addition of student choice into the ASSISTments platform [8]. This line of research examines motivation and learning when students are able to invest in the learning process. Students were randomly assigned to either Choice or No Choice conditions within a problem set on simple fraction multiplication. Those given choice were asked to select their feedback medium, while those without choice were randomly assigned to receive either text or video feedback. Results suggested that even if feedback was not ultimately used, students who were prompted to choose their feedback medium significantly outperformed those who were not. A second iteration of this study is currently underway using a new If-Then navigation infrastructure that was built because of the significant effects observed in the pilot. If previous results are replicated, these findings may be groundbreaking in that the addition of relatively inconsequential choices to adaptive tutoring systems could enhance student motivation and performance.

2.2.2 Content Delivery Patterns

Motivation and learning outcomes can also be improved by making content delivery more adaptive. Recent work within ASSISTments has revealed the benefit of interleaving (or mixing) skill content within homework settings [9]. Serving as a conceptual replication of previous work in the field, our goal was to isolate the interleaving effect within a brief homework assignment, as measured by learning gains on a delayed posttest. Using a randomized controlled trial, a practice session was presented featuring either interleaved or blocked content spanning three math skills. This study was unique in that rather than relying on a formal posttest, a second homework assignment was used to gauge learning gains through average score, hint usage, and attempt count. The use of tutoring feedback during posttest provided additional dependent variables for analysis while allowing students continued learning opportunities. Observations revealed that interleaving can be beneficial in adaptive learning environments, and appears especially significant for low performing students.

2.2.3 Assessment Enhancing Motivation

An extension of the work presented in 2.1.3, this research examined partial credit scoring using a grid search of 441 algorithmically defined models through per hint and per attempt

penalizations [6]. Binary scoring, as utilized by most adaptive tutoring systems, can serve to demotivate students from engaging with tutoring feedback and rich system features that are intended to excel beyond traditional classroom practices. For each of the 441 models examined, tables were established using maximum likelihood probabilities to predict binary next problem correctness (time $t + 1$), given the partial credit score on the current question (time t). Findings suggest that a data driven approach to defining partial credit penalization is possible and that an optimal penalization range can be isolated using model accuracy. Further, findings suggest that within the optimal range, lower penalizations do not differ significantly from higher penalizations, allowing leeway for content developers and teachers to enhance student motivation through reduced penalization.

2.3 Goals & Insight Sought

As I delve into my dissertation I expect my work to grow and meld into a unified construct surrounding the enhancement of student motivation and learning within adaptive tutoring systems. It is clear that the facets discussed here will link the two underlying realms of my research (i.e., randomized controlled trials and data mining), but it is not yet clear how. Through continued investigation of feedback, student choice, and assessment methodologies, I hope to establish a unique line of research that remains broad and yet powerful. Advice on how to drive a broad topic dissertation is sought. Essentially, I hope to gain an external expert's opinion on how to best merge the facets of my research. Advice on future endeavors within individual facets would also be appreciated.

The immediate impact of my research is already evident through continued improvements to the ASSISTments platform. The work presented here has inspired content expansion as well as infrastructure changes to enhance future research design. Within the next three years I expect that my research will continue to refine ASSISTments while increasing intellectual merit in my field. The broader impact of my work will be measured in long-term achievements that affect systemic change in education and promote data driven practices and individualized learning via adaptive tutoring platforms.

3. ACKNOWLEDGEMENTS

Funding for my work has been granted by a U.S. Department of Education GAANN fellowship (P200A120238), an Office of Naval Research grant (N00014-13-C-0127), and a PIMSE fellowship (NSF DGE-0742503). Thanks to S.O. & L.P.B.O.

4. REFERENCES

[1] Clark, R.C. & Mayer, R. E. (2003). e-Learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning. San Francisco, CA: Pfeiffer

[2] Frenzel, A.C., Pekrun, R. & Goetz, T. (2007). Girls and mathematics – A "hopeless" issue? A control-value approach to gender differences in emotions towards mathematics. Eur. J of Psych of Ed. 22 (4). pp. 497-514.

[3] Heffernan N. & Heffernan C. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. Int J Art Intel in Ed.

[4] Ostrow, K. (In Press). Motivating Learning in the Age of the Adaptive Tutor. To be included in Conati, C., Heffernan, N., Mitrovic, A., & Verdejo, M. (Eds.), Proceedings of the 17th Int Conf on AIED.

[5] Ostrow, K., Donnelly, C., Adjei, S., & Heffernan, N. (2015). Improving Student Modeling Through Partial Credit and Problem Difficulty. In Russell, Woolf, & Kiczales (Eds.), Proceedings of the 2nd ACM Conf on L@S. pp. 11-20.

[6] Ostrow, K., Donnelly, C., & Heffernan, N. (In Press). Optimizing Partial Credit Algorithms to Predict Student Performance. To be included in Romero, C., Pechenizkiy, M., Boticario, J.G., & Santos, O.C. (Eds.), Proceedings of the 8th Int Conf on EDM.

[7] Ostrow, K.S. & Heffernan, N.T. (2014). Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments. In Stamper, J., et al. (Eds) Proceedings of the 7th Int Conf on EDM. pp. 296-299.

[8] Ostrow, K. & Heffernan, N. (In Press). The Role of Student Choice Within Adaptive Tutoring. To be included in Conati, C., Heffernan, N., Mitrovic, A., & Verdejo, M. (Eds.), Proceedings of the 17th Int Conf on AIED.

[9] Ostrow, K., Heffernan, N., Heffernan, C., Peterson, Z. (In Press). Blocking vs. Interleaving: An Attempt to Replicate the Concept of Comparing Schedule Types Within One Night of Math Homework. To be included in Conati, C., Heffernan, N., Mitrovic, A., & Verdejo, M. (Eds.), Proceedings of the 17th Int Conf on AIED.

[10] Ostrow, K.S., Schultz, S.E. & Arroyo, I. (2014). Promoting Growth Mindset Within Intelligent Tutoring Systems. In CEUR-WS (1183), Gutierrez-Santos, S., & Santos, O.C. (eds) EDM 2014 Extended Proceedings: NCFPAL Workshop. pp. 88-93.

[11] Patall, E.A., Cooper, H., & Robinson, J.C. (2008). The Effects of Choice on Intrinsic Motivation and Related Outcomes: A Meta-Analysis of Research Findings. Psychology Bulletin. 134 (2), pp 270-300.

[12] Pekrun, R. (2006). The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. Educational Psychology Review, 18 (4), pp. 315-341.

[13] Razzaq, L. & Heffernan, N.T. (2006). Scaffolding vs. hints in the ASSISTments system. In Ikeda, Ashley & Chan (Eds). *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*. Springer-Verlag: Berlin. 635-644.

[14] U.S. Department of Education, Office of Educational Technology. (2010a). Transforming American Education: Learning Powered by Technology. Washington, D.C.

[15] U.S. Department of Education, Office of Planning, Evaluation, and Policy Development. (2010b). Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies. Washington, D.C.

[16] Wang, Y. & Heffernan, N. (2011). The "assistance" model: leveraging how many hints and attempts a student needs. In Proceedings of the *Florida Artificial Intelligence Research Society Conference* (FLAIRS 2011).

Estimating the Local Size and Coverage of Interaction Network Regions

Michael Eagle
North Carolina State University
Department of Computer Science
Raleigh, NC 27695-8206
mjeagle@ncsu.edu

Tiffany Barnes
North Carolina State University
Department of Computer Science
Raleigh, NC 27695-8206
tmbarnes@ncsu.edu

ABSTRACT

Interactive problem solving environments, such as intelligent tutoring systems and educational video games, produce large amounts of transactional data which make it a challenge for both researchers and educators to understand how students work within the environment. Researchers have modeled the student-tutor interactions using complex network representations to automatically derive next-step hints, derive high-level approaches, and create visualizations of student behavior. However, students do not explore the complete problem space. The nonuniform exploration of the problem results in smaller networks and less next-step hints. In this work we explore the possibility of using frequency estimation to uncover locations in the network with differing amounts of student-saturation. Identification of these regions can be used to locate specific problem approaches and strategies that would be most improved by additional student-data.

Keywords

Interaction Networks, Data-Driven, Problem Solving

1. INTRODUCTION

Data-driven methods to provide automatic hints have the potential to vastly reduce the cost associated with developing tutors with personalized feedback. Modeling the student-tutor interactions as a complex network provides a platform for researchers to generate hint-templates and automatically generate next-step hints; the interaction networks also work as useful visualization of student problem-solving, as well as a structure from which to mine high level approaches of student problem-solving approaches. Data-driven approaches require an uncertain amount of data collection before they can produce feedback, and it is not always clear how much is needed for different environments. Eagle et al. explored the structure of these student interaction networks and argued that networks could be interpreted as an empirical sample of student problem solving [4]. This would mean that students who are similar in problem-solving approaches would also be represented in the same parts of the interaction network. This would suggest that students who are more similar would have smaller networks as they explore the same parts of the problem space. We argue that as the expectation is for different populations of students to have different interaction networks and that different domains will require different amounts of student-data, there need to be good metrics for describing the quality of the networks.

In this work, we will make use of Good-Turing frequency estimation on interaction level data to predict the local size and hint-producing capability of interaction network regions. Our estimator makes use of Good-Turing frequency estimation [5]. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. In our context, the object types will refer to network-states (vertices,) and observations will refer to the student interactions (edges.)

Creation of adaptive educational programs is expensive, intelligent tutors require content experts and pedagogical experts to work with tutor developers to identify the skills students are applying and the associated feedback to deliver [7]. In order to address the difficulty in authoring intelligent tutoring content, Barnes and Stamper built an approach called the Hint Factory to use student data to build a Markov Decision Process (MDP) of student problem-solving approaches to serve as a domain model for automatic hint generation [12]. Other approaches to automated generation of feedback have attempted to condense similar solutions in order to address sparse data sets. One such approach converts solutions into a canonical form by strictly ordering the dependencies of statements in a program [9]. Another approach compares *linkage graphs* modelling how a program creates and modifies variables, with nested states created when a loop or branch appears in the code [6]. In the Andes physics tutor, students may ask for hints about how to proceed. Similarly to Hint Factory-based approaches, a solution graph representing possible correct solutions to the problem was used, however it was automatically generated rather than being derived from data, and uses plan recognition to decide which of the problem derivations the student is working towards [13].

Interaction networks are scale-free, in that there is a small

subset of the overall network-states which contain the largest number of neighboring states [4]. Eagle et al. argued that this was in part due to the nature of the problem-solving environment, where by students with similar problem solving ability and preferences would travel into similar parts of the network and problem-features would result in some states being more important to the problem than others [4]. With this interpretation as a basis sub-regions of the network corresponding to high-level approaches to the problem were shown to capture problem-solving differences between two experimental groups [3]. A region of the network representing a minority approach, would result in locations of the network that would not produce adequate hints for students taking that approach.

2. INTERACTION NETWORKS

An *Interaction Network* is a complex network representation of all observed student and tutor interactions for a given problem in a game or tutoring system [4]. To construct an Interaction Network for a problem, we collect the set of all solution attempts for that problem. Each solution attempt is defined by a unique user identifier, as well as an ordered sequence of interactions, where an interaction is defined as {initial state, action, resulting state}, from the start of the problem until the user solves the problem or exits the system. The information contained in a *state* is sufficient to precisely recreate the tutor's interface at each step. Similarly, an *action* is any user interaction which changes the state, and is defined as {action name, pre-conditions, result}. Regions of the network can be discovered by applying network clustering methods, such as those used by Eagle et al. for deriving maps high-level student approaches to problems [3].

Stamper and Barnes' Hint Factory approach generates a next-step Hint Policy by modeling student-tutor interactions as a Markov Decision Process [12]. This has been adapted to work with interaction networks by using a value-iteration algorithm [2] on the states [4]. We define a state, S to be *Hintable* if there exists a path on the network to a goal-state starting from S. We define the *Hintable* network to be the induced subset of the interaction network containing only *Hintable* states.

The "cold start problem" is an issue that arises in all data-driven systems where for early users of the system, predictions made are inaccurate or incomplete [11, 10]. Barnes and Stamper [1] approached the question of how much data is needed to get a certain amount of overlap in student solution attempts by incrementally adding student attempts and measuring the step overlap over a large series of trials. This was done with the goal of producing automatically generated hints, and thus solution-attempts that did not reach the goal were excluded. Peddycord et al. [8] performed a similar technique to evaluate differences in overlap between two different interaction network state representations.

2.1 Good-Turing Network Estimation

In this work, we are presenting a new method for estimating the size of the unobserved portion of a partially constructed Interaction Network. Our estimator makes use of Good-Turing frequency estimation [5]. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. Gale and Sampson revisited and simplified the implementation [5]. In its original context, given a sample text from a vocabulary, the Good-Turing Estimator will predict the probability that a new word selected from that vocabulary will be one not previously observed.

The Good-Turing method of estimation uses the frequency of frequencies for the sample text in order to estimate the probability that a new word will be of a given frequency. Based on this distribution, we calculate the probability of observing a new word in the vocabulary based on the observed probability of observing a word with frequency 1. Therefore, the expected probability of the next observation being an unseen word P_0 is estimated by:

$$P_0 = \frac{N_1}{N} \tag{1}$$

Where N_1 is the total number of words occurring with frequency 1, and N is the total number of observations. Since N_1 is the largest and best explored group of words, the so-far observed value of N_1 is a reasonable estimate of P_1. To apply this method to an interaction network, we will estimate the probability of encountering a new state, based on the previously seen state frequencies. P_0 can then be used to smooth the estimation proportions of the other states.

Our version of P_0 is the probability of encountering a new state (a state that currently has a frequency of zero,) on a new interaction. We also interpret this as the proportion of the network missing from the sample. We will refer to an interaction with a unobserved state as having *fallen off* of the interaction network. We will use the complement of P_0 as the estimate of *network coverage*, I_C, the probability that a new interaction will remain on the network: $I_C = 1 - P_0$.

The *state space* of the environment is the set of all possible state configurations. For both the BOTS game and the Deep Thought tutor the potential state space is infinite. For example, in the Deep Thought tutor a student can always use the addition rule to add new propositions to the state. However, as argued in Eagle et. al. [4], the actions that reasonable humans perform is only a small subset of the theoretical state space; the actions can also be different for different populations of humans. We will refer to this subset as the *Reasonable State Space*, with *unreasonable* being loosely defined as actions that we would not expect a human to take. An interaction network is an empirical sample of the problem solving behavior from a particular population, and is a subset of the state space of all possible *reasonable* behaviors. Therefore, our metrics P_0 and I_C are estimates of how well the observed interaction network represents the reasonable state space.

3. DISCUSSION

Figure 1 shows the results of a preliminary analysis on an interaction network based on student-log data from a tutoring environment. For each region we calculated values of network coverage, I_C, and have highlighted regions of the network which have values below 90% coverage. Good-

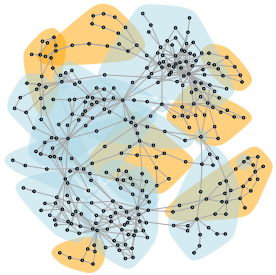


Figure 1: An interaction network with regions of high coverage highlighted in light blue and regions of low coverage highlighted in orange. The low coverage regions of the network require more data before coverage could reach a I_C level above 90%.

Turing Estimation works well in the contexts of interaction networks. Our network coverage metric I_C allows a quick and easy to calculate method of comparing different state representations, as well as quantifying the difference. New methods for improving automatic hint generation can target these areas of the network which have the lowest coverage, such as asking for instructor input on specific regions or by starting advanced students in these regions in order to observe their paths out.

We were also able to interpret this metric as measure of the proportion of the network not yet observed P_0. On a high-level this value alone is a useful metric for the percentage of times a student-interaction is to a not yet observed state. The P_0 score for the hint-able network is likewise a measure for the probability that a student will "fall off" of the network from which we can provide feedback. Therefore, we can use the P_0 metric to predict next-step "fall off" we could estimate the "risk" of different network regions. If we are reasonably sure that the majority of successful paths to the goal have been previously observed then falling off of the network likely means that the student is unlikely to reach the goal.

Region-level coverage also has implications given our previous theories on the network being a sample created from bias (non-random) walks on the problem-space, as the more homogeneous the bias-walkers are, the faster the network will represent the population and smaller total states explored will be. We revisited the results of [3], and have added more description to the effect of hint; students with access to hints explored less overall unique states which implies that the students were more similar to each other in terms of the types of actions and states they visited within the problem.

Future directions for this research include general improvements to the network clustering algorithms which generate the regions. Regions which have low coverage might not be worth separating from their parent region for visualization or high-level hint generation processes. The local and global measures of network coverage can help identify problematic regions in interaction networks which could harm hint production; they also provide a metric to evaluate new, "cold start" problems and make sure that enough data has been collected in order produce hints to multiple problem solving approaches. Finally, exploration of coverage between groups has the potential to uncover differences in problem solving behavior, and improve automatic hinting and understanding of student approaches to problems.

4. REFERENCES

[1] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)*, pages 373–382, 2008.

[2] R. Bellman. A markovian decision process. Technical report, DTIC Document, 1957.

[3] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. *Proceedings of the Seventh International Conference on Educational Data Mining*, 2014.

[4] M. Eagle, D. Hicks, P. III, and T. Barnes. Exploring networks of problem-solving interactions. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.

[5] W. A. Gale and G. Sampson. Good-turing frequency estimation without tears*. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

[6] W. Jin, T. Barnes, J. Stamper, M. J. Eagle, M. W. Johnson, and L. Lehmann. Program representation for automatic hint generation for a data-driven novice programming tutor. In *Intelligent Tutoring Systems*, pages 304–309. Springer, 2012.

[7] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[8] B. Peddycord III, A. Hicks, and T. Barnes. Generating hints for programming problems using intermediate output.

[9] K. Rivers and K. R. Koedinger. Automating hint generation with solution space path construction. In *Intelligent Tutoring Systems*, pages 329–339. Springer, 2014.

[10] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[11] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[12] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2013.

[13] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.